



# **MSR Engineering Documentation (MEDOC) MSRSW.DTD Elements and attributes**

**Scope: Software for electronic control units**

**MSR-MEDOC, Dipl.-Ing. Bernhard Weichel**



## Abstract

This is a reference document for *MSRSW DTD* describing all elements.  
*MSRSW DTD* is part of the MSR development documentation MEDOC.



# Table of Contents

	Table of Contents	3
	Introduction	11
1	<b>How to read this document</b>	<b>12</b>
1.1	The following conventions apply	12
1.2	Graphical conventions used in DTD diagrams	12
2	<b>Allgemeine Projektdaten</b>	<b>14</b>
1	<b>ABS ... ARRAYSIZE</b>	<b>15</b>
1.1	ABS	15
1.2	ACCEPTANCE-COND	15
1.3	ADD-INFO	15
1.4	ADD-SPEC	16
1.5	ADDRESS	17
1.6	ADMIN-DATA	17
1.7	ANNOTATION	18
1.8	ANNOTATION-ORIGIN	19
1.9	ANNOTATION-TEXT	19
1.10	ANNOTATIONS	19
1.11	ARRAYSIZE	20
2	<b>BIT-BASE-NAME ... BIT-POS</b>	<b>21</b>
2.1	BIT-BASE-NAME	21
2.2	BIT-BASE-TYPE	21
2.3	BIT-COUNT	21
2.4	BIT-POS	22
3	<b>C-CODE ... COND</b>	<b>23</b>
3.1	C-CODE	23
3.2	CHANGE	23
3.3	CHAPTER	23
3.4	CITY	24
3.5	CMT-INT	25
3.6	CMT-PHYS	25
3.7	CMT-TEXT	25
3.8	CODE	26
3.9	CODED	26
3.10	CODED-MAX	26
3.11	CODED-MIN	27
3.12	COLSPEC	27



3.13	COMPANIES	28
3.14	COMPANY	28
3.15	COMPANY-DOC-INFO	29
3.16	COMPANY-DOC-INFOS	30
3.17	COMPANY-REF	30
3.18	COMPANY-REVISION-INFO	31
3.19	COMPANY-REVISION-INFOS	31
3.20	COND	32
4	<b>COUNT ... COUNT</b>	<b>33</b>
4.1	COUNT	33
5	<b>DATE ... DOC-REVISIONS</b>	<b>34</b>
5.1	DATE	34
5.2	DATE-1	34
5.3	DEF	34
5.4	DEF-ITEM	35
5.5	DEF-LIST	35
5.6	DEMARICATION-OTHER-PROJECTS	35
5.7	DEPARTMENT	36
5.8	DESC	36
5.9	DIR-HAND-OVER-DOC-DATA	37
5.10	DOC-LABEL	37
5.11	DOC-REVISION	37
5.12	DOC-REVISIONS	38
6	<b>E ... ENTRY</b>	<b>39</b>
6.1	E	39
6.2	EMAIL	39
6.3	ENTITY-NAME	39
6.4	ENTRY	39
7	<b>FAX ... FUNCTION-REF</b>	<b>41</b>
7.1	FAX	41
7.2	FIGURE	41
7.3	FILE	41
7.4	FORMULA	42
7.5	FT	42
7.6	FUNCTION-REF	43
8	<b>GENERAL-PROJECT-DATA ... GRAPHIC</b>	<b>44</b>
8.1	GENERAL-PROJECT-DATA	44
8.2	GENERIC-MATH	44
8.3	GRAPHIC	45



9	<b>HOMEPAGE ... HOMEPAGE</b>	<b>47</b>
9.1	HOMEPAGE	47
10	<b>IE ... ITEM-LABEL</b>	<b>48</b>
10.1	IE	48
10.2	INDENT-SAMPLE	48
10.3	INTEGRATION-CAPABILITY	48
10.4	INTRODUCTION	49
10.5	ITEM	50
10.6	ITEM-LABEL	50
11	<b>LABEL ... LONG-NAME-1</b>	<b>52</b>
11.1	LABEL	52
11.2	LABELED-ITEM	52
11.3	LABELED-LIST	53
11.4	LABELS	53
11.5	LANGUAGE	54
11.6	LIST	54
11.7	LOCS	55
11.8	LONG-NAME	55
11.9	LONG-NAME-1	56
12	<b>MAX ... MSRSW</b>	<b>57</b>
12.1	MAX	57
12.2	MAX-COUNT	57
12.3	MIN	57
12.4	MODIFICATION	58
12.5	MODIFICATIONS	58
12.6	MSRSW	59
13	<b>NA ... NUMBER</b>	<b>61</b>
13.1	NA	61
13.2	NAMELOC	61
13.3	NCOI-1	61
13.4	NMLIST	62
13.5	NOTE	63
13.6	NUMBER	63
14	<b>OBJECTIVES ... OVERALL-PROJECT</b>	<b>64</b>
14.1	OBJECTIVES	64
14.2	OFFSET	64
14.3	OVERALL-PROJECT	64
15	<b>P ... PURCHASING-COND</b>	<b>66</b>
15.1	P	66



15.2	PARALLEL-DESIGNS	66
15.3	PHONE	67
15.4	PHYS	67
15.5	PHYS-MAX	67
15.6	PHYS-MIN	67
15.7	POSITION	68
15.8	PRIVATE-CODE	68
15.9	PRIVATE-CODES	68
15.10	PRM	69
15.11	PRM-CHAR	69
15.12	PRMS	70
15.13	PROG-CODE	71
15.14	PROJECT	72
15.15	PROJECT-DATA	72
15.16	PROJECT-SCHEDULE	72
15.17	PROTOCOLS	73
15.18	PUBLISHER	73
15.19	PURCHASING-COND	73
16	<b>REASON ... ROW</b>	<b>75</b>
16.1	REASON	75
16.2	REASON-ORDER	75
16.3	REMARK	75
16.4	REVISION-LABEL	76
16.5	ROLE	76
16.6	ROLES	77
16.7	ROW	77
17	<b>SAMPLE ... SW-ADDRESSING-METHODS</b>	<b>78</b>
17.1	SAMPLE	78
17.2	SAMPLE-REF	78
17.3	SAMPLE-SPEC	78
17.4	SAMPLES	79
17.5	SCHEDULE	79
17.6	SHIFT	79
17.7	SHORT-NAME	80
17.8	SI-UNIT	80
17.9	SPANSPEC	81
17.10	STATE	81
17.11	STATE-1	82
17.12	STD	82
17.13	SUB	83
17.14	SUBTITLE	83
17.15	SUP	83



17.16	SW-ADDRESSING-METHOD	84
17.17	SW-ADDRESSING-METHOD-CLASS	84
17.18	SW-ADDRESSING-METHOD-DESC	85
17.19	SW-ADDRESSING-METHOD-REF	85
17.20	SW-ADDRESSING-METHODS	86
18	<b>SW-APPLICATION-NOTES ... SW-COMPU-METHOD-TEXT-PAIR</b>	<b>87</b>
18.1	SW-APPLICATION-NOTES	87
18.2	SW-ARRAY-INDEX	87
18.3	SW-ASAP-6-PRM-METHOD	87
18.4	SW-AXIS-GROUPED	88
18.5	SW-AXIS-INDIVIDUAL	88
18.6	SW-AXIS-SHIFT-OFFSET	89
18.7	SW-BASE-TYPE	89
18.8	SW-BIT-REPRESENTATION	89
18.9	SW-CARB-DOC	90
18.10	SW-CODE-SYNTAX	90
18.11	SW-CODE-SYNTAX-CLASS	91
18.12	SW-CODE-SYNTAX-DESC	92
18.13	SW-CODE-SYNTAX-REF	92
18.14	SW-CODE-SYNTAXES	93
18.15	SW-COMPU-GENERIC-MATH	93
18.16	SW-COMPU-METHOD	94
18.17	SW-COMPU-METHOD-REF	94
18.18	SW-COMPU-METHOD-TABLE	95
18.19	SW-COMPU-METHOD-TEXT	95
18.20	SW-COMPU-METHOD-TEXT-PAIR	96
19	<b>SW-COMPU-METHOD-VALUE-PAIR ... SW-FUNCTION-SPEC</b>	<b>97</b>
19.1	SW-COMPU-METHOD-VALUE-PAIR	97
19.2	SW-COMPU-METHODS	97
19.3	SW-DATA-DICTIONARIES	97
19.4	SW-DATA-DICTIONARY	98
19.5	SW-DATA-DICTIONARY-SPEC	99
19.6	SW-DATA-TYPE-SCALING	100
19.7	SW-DATADICTIONARY-CLASS	101
19.8	SW-DECOMPOSITION	101
19.9	SW-FULFILS	101
19.10	SW-FUNCTION	101
19.11	SW-FUNCTION-CLASS	102
19.12	SW-FUNCTION-DEF	102
19.13	SW-FUNCTION-DESC	103
19.14	SW-FUNCTION-EXPORT-VARIABLES	103
19.15	SW-FUNCTION-IMPORT-VARIABLES	104



19.16	SW-FUNCTION-LOCAL-VARIABLES	104
19.17	SW-FUNCTION-MODELONLY-VARIABLES	105
19.18	SW-FUNCTION-REF	105
19.19	SW-FUNCTION-REFS	105
19.20	SW-FUNCTION-SPEC	106
20	<b>SW-FUNCTION-VARIABLES ... SW-PARAM-CONTENT-Y</b>	<b>107</b>
20.1	SW-FUNCTION-VARIABLES	107
20.2	SW-FUNCTION-VARIANT	107
20.3	SW-FUNCTION-VARIANTS	108
20.4	SW-FUNCTIONS	109
20.5	SW-GLOSSARY	110
20.6	SW-INTERPOLATION-METHOD	110
20.7	SW-LIMITS	110
20.8	SW-MAINTENANCE-NOTES	111
20.9	SW-PARAM	112
20.10	SW-PARAM-ARRAY-X	113
20.11	SW-PARAM-ARRAY-XY	113
20.12	SW-PARAM-AXIS-VALUES	114
20.13	SW-PARAM-AXIS-X	114
20.14	SW-PARAM-AXIS-Y	115
20.15	SW-PARAM-CLASS	115
20.16	SW-PARAM-CONTENT	116
20.17	SW-PARAM-CONTENT-TEXT	117
20.18	SW-PARAM-CONTENT-V	117
20.19	SW-PARAM-CONTENT-X	118
20.20	SW-PARAM-CONTENT-Y	118
21	<b>SW-PARAM-CONTENTS ... SW-PARAM-VALUES-GENERIC</b>	<b>119</b>
21.1	SW-PARAM-CONTENTS	119
21.2	SW-PARAM-CONTENTS-CLASS	119
21.3	SW-PARAM-CONTENTS-SPEC	119
21.4	SW-PARAM-GROUP-AXIS	120
21.5	SW-PARAM-NOEFFECT-VALUE	120
21.6	SW-PARAM-RECORD-LAYOUT	121
21.7	SW-PARAM-RECORD-LAYOUT-CLASS	121
21.8	SW-PARAM-RECORD-LAYOUT-DESC	122
21.9	SW-PARAM-RECORD-LAYOUT-REF	122
21.10	SW-PARAM-RECORD-LAYOUTS	123
21.11	SW-PARAM-REF	123
21.12	SW-PARAM-REFS	123
21.13	SW-PARAM-SINGLE-VALUE	124
21.14	SW-PARAM-TARGET	124
21.15	SW-PARAM-TEXT	124



21.16	SW-PARAM-VALUE-BLOCK	125
21.17	SW-PARAM-VALUES-ADR	125
21.18	SW-PARAM-VALUES-CODED	125
21.19	SW-PARAM-VALUES-CODED-HEX	126
21.20	SW-PARAM-VALUES-GENERIC	126
22	<b>SW-PARAM-VALUES-PHYS ... SW-VARIABLE-REF</b>	<b>127</b>
22.1	SW-PARAM-VALUES-PHYS	127
22.2	SW-PARAMS	127
22.3	SW-PHYSIC-TYPE	127
22.4	SW-PHYSIC-TYPE-1	128
22.5	SW-PHYSIC-TYPE-REF	129
22.6	SW-PHYSIC-TYPES	129
22.7	SW-RESOLUTION	130
22.8	SW-TEST-SPEC	130
22.9	SW-UNIT	130
22.10	SW-UNIT-DISPLAY	131
22.11	SW-UNIT-FROM-REF-METHOD	131
22.12	SW-UNIT-REF	132
22.13	SW-UNIT-TO-REF-METHOD	132
22.14	SW-UNITS	132
22.15	SW-VAR-INIT-VALUE	133
22.16	SW-VAR-NOT-AV-VALUE	133
22.17	SW-VARIABLE	134
22.18	SW-VARIABLE-IMPLEMENTATION	135
22.19	SW-VARIABLE-KIND	135
22.20	SW-VARIABLE-REF	136
23	<b>SW-VARIABLE-SAMPLE-RATE ... SYSTEM-OVERVIEW</b>	<b>137</b>
23.1	SW-VARIABLE-SAMPLE-RATE	137
23.2	SW-VARIABLES	137
23.3	SW-VARIABLES-READ	137
23.4	SW-VARIABLES-READWRITE	138
23.5	SW-VARIABLES-WRITE	138
23.6	SYSTEM-OVERVIEW	138
24	<b>TABLE ... TYP</b>	<b>140</b>
24.1	TABLE	140
24.2	TBD	140
24.3	TBODY	141
24.4	TBR	141
24.5	TEAM-MEMBER	142
24.6	TEAM-MEMBER-REF	143
24.7	TEAM-MEMBER-REFS	143
24.8	TEAM-MEMBERS	143



24.9	TEX-MATH	144
24.10	TEXT	144
24.11	TFOOT	145
24.12	TGROUP	145
24.13	THEAD	146
24.14	TOL	146
24.15	TOPIC-1	146
24.16	TOPIC-2	147
24.17	TT	148
24.18	TYP	148
25	<b>UNIT ... USED-LANGUAGES</b>	<b>149</b>
25.1	UNIT	149
25.2	USED-LANGUAGES	149
26	<b>V ... VERBATIM</b>	<b>150</b>
26.1	V	150
26.2	VALUE	150
26.3	VARIANT-CHAR	150
26.4	VARIANT-CHAR-ASSIGN	151
26.5	VARIANT-CHAR-ASSIGNS	151
26.6	VARIANT-CHAR-REF	151
26.7	VARIANT-CHAR-VALUE	152
26.8	VARIANT-CHARS	152
26.9	VARIANT-DEF	152
26.10	VARIANT-DEF-REF	153
26.11	VARIANT-DEF-REFS	153
26.12	VARIANT-DEFS	153
26.13	VARIANT-SPEC	154
26.14	VERBATIM	154
27	<b>XDOC ... XREF</b>	<b>155</b>
27.1	XDOC	155
27.2	XFILE	155
27.3	XREF	155
28	<b>ZIP ... ZIP</b>	<b>157</b>
28.1	ZIP	157
	Documentadministration	158



## Introduction

*Companies* **MSR-MEDOC [MSR-MEDOC]**

Name Roles	Departement	Address	Contact
Dipl.-Ing.(FH) Uwe B- less			
Dipl.-Inform. Helmut Gengenbach			
Dipl. Ing. Eckard Jakobi			
Dipl.-Ing. Herbert K- lein			
Dipl. Ing. Oliver Mar- cks			
Dipl.-Inform. Peter Rauleder			
Dipl.-Ing. Martin Trin- schek			
Dipl.-Ing. Bernhard Weichel			

*Version Information*

Document Part	Editor			
	Company	Version	State	Remarks
19.2.1999	Dipl.-Ing. Bernhard Weichel			
<a href="#">For details refer to nr. 1, Page 158</a>	MSR-MEDOC	1	rd	

# 1 How to read this document

## 1.1 The following conventions apply

This document is written using *MSRREP DTD*. The following conventions apply to this document:

- <msrsw>** SGML elements are noted as technical term **[type]=SGMLTAG**.
- [type]** SGML attributes are noted as technical term **[type]=SGML-attribute**.
- sgml-attribute* Values of SGML attributes or discrete values for elements are noted as technical term **[type]=code**
- ASAP2* The considered languages resp. DTDs are marked as technical term **[type]=product**.
- ASAP* The committees are noted as **[type]=organization**
- ECU* Objects in general are marked as technical terms **[type]=other**. This might be automotive equipments general objects such as variables etc.

## 1.2 Graphical conventions used in DTD diagrams

The structure of DTDs is shown in the MSR document as DTD diagrams (see [Figure 1 convention in DTD diagrams p. 12](#)).

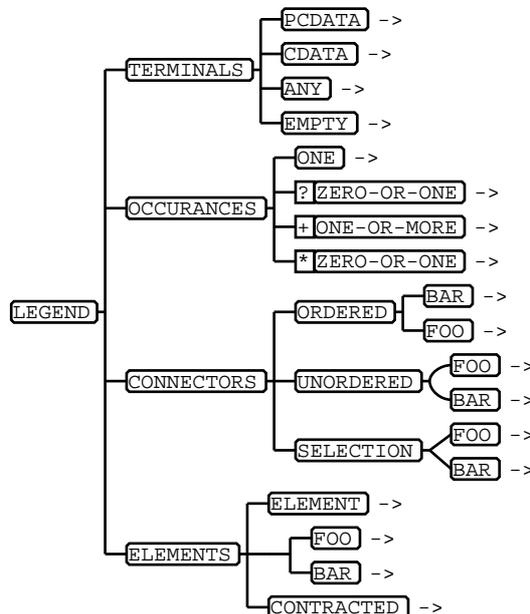


Figure 1: convention in DTD diagrams

The meaning of the symbols is:

- PCDATA** **Processable Character Data (PCDATA)** Data that consists of zero or more characters of both text and markup. PCDATA is a declared content keyword. PCDATA is used to indicate that all markup delimiters defined in the SGML declaration will be recognized by the parser as markup in the given element rather than data characters.



RCDATA	<b>Replaceable Character Data (RCDATA)</b> is data that consists of zero or more characters, in which references to substitutions are not recognized (i.e. RCDATA may contain text and entity references, but no sub-elements). See also: CDATA PCDATA.
CDATA	<b>Character Data (CDATA)</b> consists of zero or more text characters, where no markup of any kind is recognized. CDATA is an SGML term. Note that character references are allowed in a CDATA entity (substitution) but not in CDATA content.
ANY	a terminal type indicating that the object may contain text or any element defined in the model.
EMPTY	a terminal type keyword used to indicate that there is no data (i.e. no content, sub-elements or end-tags) for the object allowed in the document instance. This keyword is often used to describe elements that are placeholders or are pointers to external or system-generated data.
One	indicates that the element or the element group occurs exactly once
ZERO-OR-ONE	indicates that the element or the element group is optional
ONE-OR-MORE	indicates that the element or the element group occurs multiple times but at least once
ZERO-OR-MORE	indicates that the element or the element group occurs multiple times but also can be missed (optional)
ORDERED	a connector used to specify that the sibling objects must appear in the document in the order shown in the model
UNORDERED	a connector used to specify that the sibling objects can appear in any order in the document.
SELECTION	a connector used to specify that only one of the sibling objects can appear in the document.
ELEMENT	indicates a single SGML structure element
COLLAPSED	indicates, that the content of the element is not displayed here



## 2 Allgemeine Projektdaten

# 1 ABS ... ARRAYSIZE

## 1.1 ABS

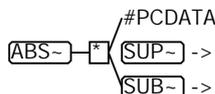


Figure 2: DTD-diagram for ABS

Child elements **<sup>** **<sub>**

parent elements **<prm-char>**

Table 1: Attributes for ABS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Absolute value for parameter characteristics. See parameter model (**<sw-prm>**).

Example

## 1.2 ACCEPTANCE-COND

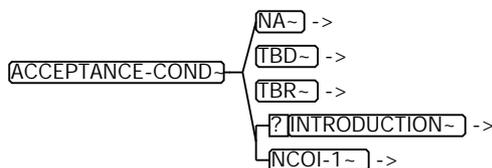


Figure 3: DTD-diagram for ACCEPTANCE-COND

Child elements **<na>** **<tbd>** **<tbr>** **<introduction>** **<ncoi-1>**

parent elements **<general-project-data>**

Table 2: Attributes for ACCEPTANCE-COND

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 1.3 ADD-INFO

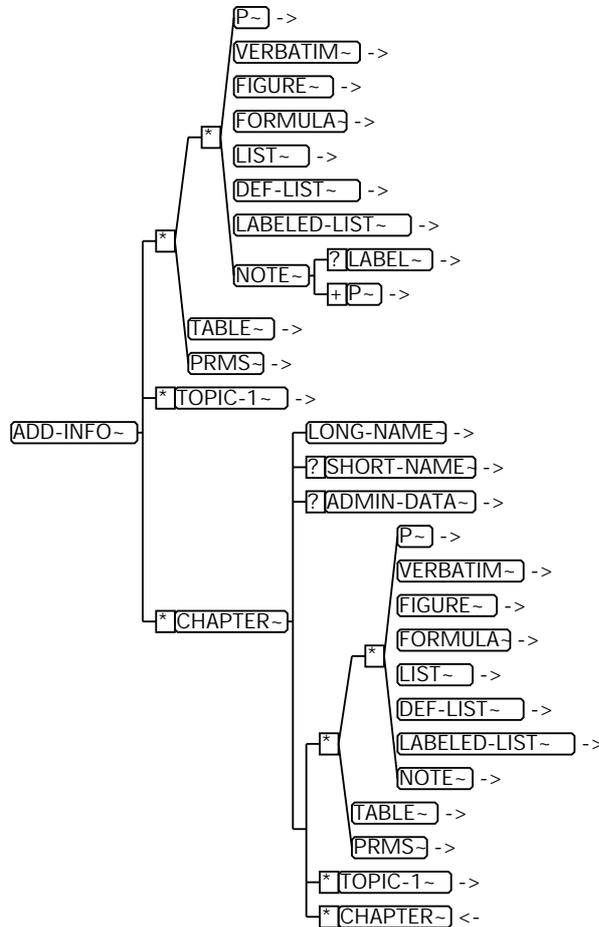


Figure 4: DTD-diagram for ADD-INFO

Child elements **<p>** **<verbatim>** **<figure>** **<formula>** **<list>** **<def-list>** **<labeled-list>** **<note>**  
**<table>** **<prms>** **<topic-1>** **<chapter>**

parent elements **<sw-function-variant>** **<sw-param>** **<sw-variable-implementation>**

Table 3: Attributes for ADD-INFO

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Additional information (which is not covered by the existing structure) can be given in this element. In opposite to **<add-spec>**, **<add-info>** complements an existing structure, while **<add-spec>** is used where no structure exists at all.

Example

## 1.4 ADD-SPEC

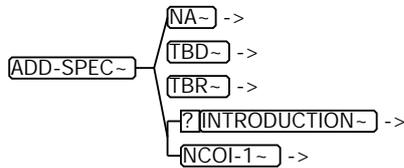


Figure 5: DTD-diagram for ADD-SPEC

Child elements `<na>` `<tdb>` `<tbr>` `<introduction>` `<ncoi-1>`

parent elements `<general-project-data>` `<msrsw>`

Table 4: Attributes for ADD-SPEC

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description This element allows to give additional specifications for which no explicit structure exists. This element usually enforces another chapter in printed material. In opposite to `<add-info>`, `<add-spec>` is used as a substitute if no structure exists for the topic, while `<add-info>` is used where an existing structure is not appropriate.

Example

## 1.5 ADDRESS

`ADDRESS~`—#PCDATA

Figure 6: DTD-diagram for ADDRESS

Child elements none

parent elements `<team-member>`

Table 5: Attributes for ADDRESS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 1.6 ADMIN-DATA

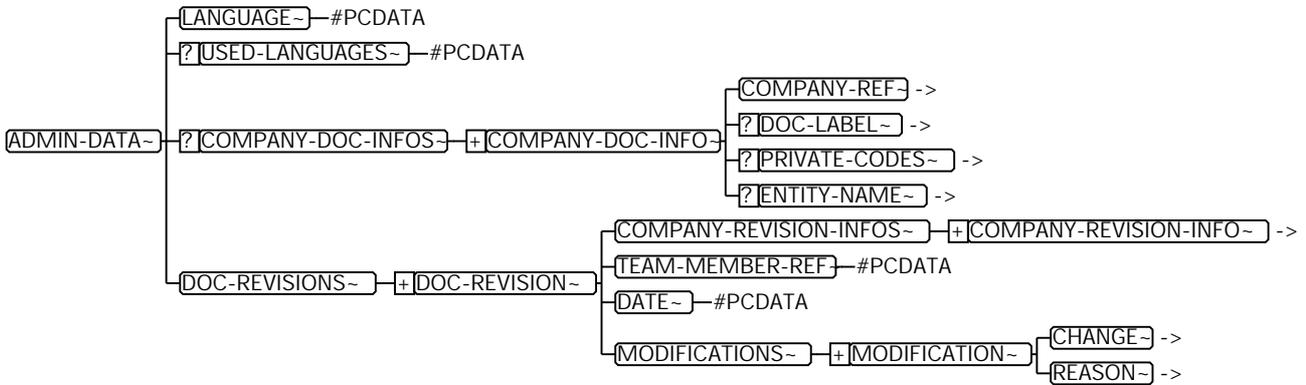


Figure 7: DTD-diagram for ADMIN-DATA

Child elements **<language>** **<used-languages>** **<company-doc-infos>** **<doc-revisions>**

parent elements **<chapter>** **<general-project-data>** **<msrsw>** **<sw-addressing-method>**  
**<sw-addressing-methods>** **<sw-code-syntax>** **<sw-code-syntaxes>**  
**<sw-compu-method>** **<sw-compu-methods>** **<sw-data-dictionary>** **<sw-**  
**data-dictionary-spec>** **<sw-function>** **<sw-function-spec>** **<sw-function-**  
**variant>** **<sw-glossary>** **<sw-param>** **<sw-param-contents>** **<sw-param-**  
**contents-spec>** **<sw-param-record-layout>** **<sw-param-record-layouts>**  
**<sw-params>** **<sw-physic-type>** **<sw-physic-types>** **<sw-unit>** **<sw-units>**  
**<sw-variable>** **<sw-variables>**

Table 6: Attributes for ADMIN-DATA

Name	Type	Class	Value	Remark
[F-CHILD-TYPE]	cdata	fixed	language:selection	
[S]	cdata	implied		

Description This element allows to specify administrative information regarding the substructure defined by the parent. This information may be versioning, fragmentation etc. The information rather covers the administration of the SGML-instance than the administration of the described subjects.

Example

## 1.7 ANNOTATION

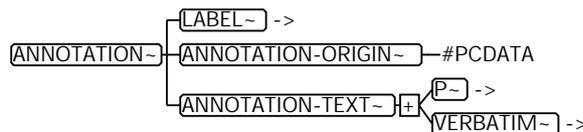


Figure 8: DTD-diagram for ANNOTATION

Child elements **<label>** **<annotation-origin>** **<annotation-text>**

parent elements **<annotations>**

**Table 7: Attributes for ANNOTATION**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Information of different kinds can arise throughout the course of a process, (e.g. processing information, notes etc.) that shall be linked to the variables, parameters or function and be forwarded in the process. Such kinds of information are stored in **<annotation>**.

Example

## 1.8 ANNOTATION-ORIGIN

**ANNOTATION-ORIGIN~** -#PCDATA

**Figure 9: DTD-diagram for ANNOTATION-ORIGIN**

Child elements none

parent elements **<annotation>**

**Table 8: Attributes for ANNOTATION-ORIGIN**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 1.9 ANNOTATION-TEXT

**ANNOTATION-TEXT~** + { **P~** -> **VERBATIM~** -> }

**Figure 10: DTD-diagram for ANNOTATION-TEXT**

Child elements **<p>** **<verbatim>**

parent elements **<annotation>**

**Table 9: Attributes for ANNOTATION-TEXT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 1.10 ANNOTATIONS

**ANNOTATIONS~** + **ANNOTATION~** { **LABEL~** -> **ANNOTATION-ORIGIN~** -> **ANNOTATION-TEXT~** -> }

**Figure 11: DTD-diagram for ANNOTATIONS**

Child elements **<annotation>**



parent elements **<sw-function>** **<sw-function-variant>** **<sw-param>** **<sw-param-content>**  
**<sw-param-contents>** **<sw-variable-implementation>**

**Table 10: Attributes for ANNOTATIONS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 1.11 ARRAYSIZE

`ARRAYSIZE~`—#PCDATA

**Figure 12: DTD-diagram for ARRAYSIZE**

Child elements none

parent elements **<sw-param>** **<sw-variable-implementation>**

**Table 11: Attributes for ARRAYSIZE**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Variables **<sw-variable>** as well as calibration parameters **<sw-param>** may exist as an array. In this case the dimension of the array is given in **<arraysize>**. Dimensions are integer numbers and can be separated by blanks in case of multi-dimensional arrays.

Example

## 2 BIT-BASE-NAME ... BIT-POS

### 2.1 BIT-BASE-NAME



Figure 13: DTD-diagram for BIT-BASE-NAME

Child elements none

parent elements **<sw-bit-representation>**

Table 12: Attributes for BIT-BASE-NAME

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description If a variable is a set of bits, there is another variable hosting the single bits. **<bit-base-name>** gives the name of this "host" variable.

Example

### 2.2 BIT-BASE-TYPE



Figure 14: DTD-diagram for BIT-BASE-TYPE

Child elements none

parent elements **<sw-bit-representation>**

Table 13: Attributes for BIT-BASE-TYPE

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description If a variable is a set of bits, there is another variable hosting the single bits. **<bit-base-type>** gives the base type of this "host" variable (e.g. *ubyte*). The value must be agreed between project partners.

Example

### 2.3 BIT-COUNT



Figure 15: DTD-diagram for BIT-COUNT

Child elements none

parent elements **<sw-bit-representation>**



**Table 14: Attributes for BIT-COUNT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description If a variable is a set of bits, there is another variable hosting the single bits. **<bit-count>** gives the number of bits allocated in the "host" variable .

Example

## 2.4

## BIT-POS

#PCDATA

**Figure 16: DTD-diagram for BIT-POS**

Child elements none

parent elements **<sw-bit-representation>**

**Table 15: Attributes for BIT-POS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description If a variable is a set of bits, there is another variable hosting the single bits. **<bit-pos>** gives the the position of the first bit in the "host" variable.

Example

### 3 C-CODE ... COND

#### 3.1 C-CODE

`<C-CODE>`—#PCDATA

Figure 17: DTD-diagram for C-CODE

Child elements none

parent elements `<formula>`

Table 16: Attributes for C-CODE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

#### 3.2 CHANGE

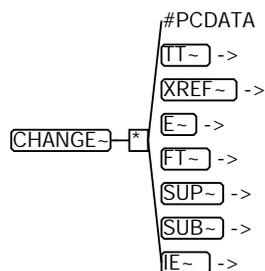


Figure 18: DTD-diagram for CHANGE

Child elements `<tt>` `<xref>` `<e>` `<ft>` `<sup>` `<sub>` `<ie>`

parent elements `<modification>`

Table 17: Attributes for CHANGE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description More or less detailed description of the performed changes can be given here.

Example

### 3.3 CHAPTER

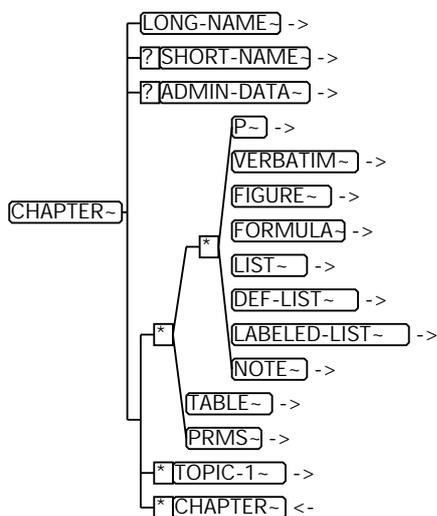


Figure 19: DTD-diagram for CHAPTER

Child elements **<long-name>** **<short-name>** **<admin-data>** **<p>** **<verbatim>** **<figure>** **<formula>** **<list>** **<def-list>** **<labeled-list>** **<note>** **<table>** **<prms>** **<topic-1>** **<chapter>**

parent elements **<add-info>** **<chapter>** **<ncoi-1>** **<sw-addressing-method-desc>** **<sw-application-notes>** **<sw-carb-doc>** **<sw-code-syntax-desc>** **<sw-function-desc>** **<sw-maintenance-notes>** **<sw-param-record-layout-desc>** **<sw-test-spec>**

Table 18: Attributes for CHAPTER

Name	Type	Class	Value	Remark
<b>[BREAK]</b>	nmtkgrp	implied	BREAK NO-BREAK	
<b>[F-ID-CLASS]</b>	name	fixed	CHAPTER	
<b>[HELP-ENTRY]</b>	cdata	implied		
<b>[ID]</b>	id	required		
<b>[S]</b>	cdata	implied		

Description **<chapter>** is used to put a presentation structure on prose descriptions. Chapters can be structured hierachically.

Example

### 3.4 CITY

**CITY~**—#PCDATA

Figure 20: DTD-diagram for CITY

Child elements none

parent elements **<team-member>**

**Table 19: Attributes for CITY**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

### 3.5

## CMT-INT

`<CMT-INT~>#PCDATA`

**Figure 21: DTD-diagram for CMT-INT**

Child elements none

parent elements `<sw-compu-method-text-pair>` `<sw-compu-method-value-pair>`

**Table 20: Attributes for CMT-INT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Internal value for a conversion formula

Example

### 3.6

## CMT-PHYS

`<CMT-PHYS~>#PCDATA`

**Figure 22: DTD-diagram for CMT-PHYS**

Child elements none

parent elements `<sw-compu-method-value-pair>`

**Table 21: Attributes for CMT-PHYS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Physical value for a conversion formula defined as conversion table.

Example

### 3.7

## CMT-TEXT

`<CMT-TEXT~>#PCDATA`

**Figure 23: DTD-diagram for CMT-TEXT**

Child elements none

parent elements `<sw-compu-method-text-pair>`

**Table 22: Attributes for CMT-TEXT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Value in text form for a conversion formula.

Example

## 3.8 CODE

`[CODE-]`—#PCDATA

**Figure 24: DTD-diagram for CODE**

Child elements none

parent elements `<variant-char>` `<variant-char-value>` `<variant-def>`

**Table 23: Attributes for CODE**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description The definition of variants end up in code names for variants resp. for variant characteristics resp. for values of variant characteristics. `<code>` takes these names. This code may be a number as well as a name.

Example

## 3.9 CODED

`[CODED-]`

- `[CODED-MIN-]`—#PCDATA
- `[CODED-MAX-]`—#PCDATA

**Figure 25: DTD-diagram for CODED**

Child elements `<coded-min>` `<coded-max>`

parent elements `<sw-limits>` `<sw-limits>`

**Table 24: Attributes for CODED**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Coded/internal values for limits.

Example

### 3.10 CODED-MAX

`<CODED-MAX>`—#PCDATA

Figure 26: DTD-diagram for CODED-MAX

Child elements none

parent elements `<coded>`

Table 25: Attributes for CODED-MAX

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Maximum coded limiting value

Example

### 3.11 CODED-MIN

`<CODED-MIN~>`—#PCDATA

Figure 27: DTD-diagram for CODED-MIN

Child elements none

parent elements `<coded>`

Table 26: Attributes for CODED-MIN

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Minimum coded limiting value

Example

### 3.12 COLSPEC

`<COLSPEC>`empty

Figure 28: DTD-diagram for COLSPEC

Child elements none

parent elements `<tfoot>` `<tgroup>` `<thead>`

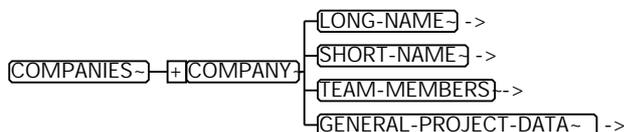
Table 27: Attributes for COLSPEC

Name	Type	Class	Value	Remark
[ALIGN]	nmtkgrp	implied	LEFT RIGHT CENTER JUSTIFY CHAR	
[CHAR]	cdata	implied		

**Table 27 (Cont.): Attributes for COLSPEC**

Name	Type	Class	Value	Remark
[CHAROFF]	nutoken	implied		
[COLNAME]	nmtoken	implied		
[COLNUM]	number	implied		
[COLSEP]	number	implied		
[COLWIDTH]	cdata	implied		
[ROWSEP]	number	implied		
[S]	cdata	implied		

### 3.13 COMPANIES



**Figure 29: DTD-diagram for COMPANIES**

Child elements **<company>**

parent elements **<project>**

**Table 28: Attributes for COMPANIES**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Gets information about all companies participating in the project in question.

Example

### 3.14 COMPANY

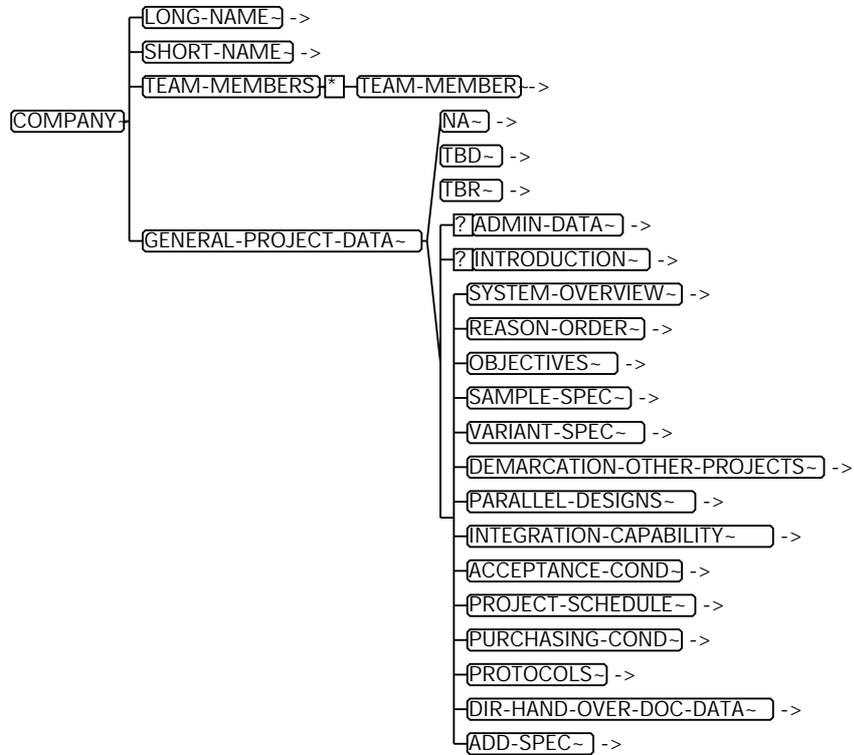


Figure 30: DTD-diagram for COMPANY

Child elements <long-name> <short-name> <team-members> <general-project-data>  
parent elements <companies>

Table 29: Attributes for COMPANY

Name	Type	Class	Value	Remark
[F-CHILD-TYPE]	cdata	fixed	long-name:selection	
[F-ID-CLASS]	name	fixed	COMPANY	
[F-NAMESPACE]	names	fixed	SAMPLE TEAM-MEMBER VARIANT-DEF VARIANT-CHAR	
[ID]	id	required		
[ROLE]	nmtkgrp	required	MANUFACTURER SUPPLIER	
[S]	cdata	implied		

Description Company-specific details for a company participating in the project

Example

### 3.15 COMPANY-DOC-INFO

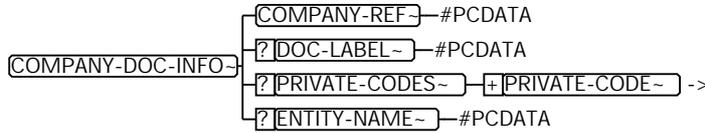


Figure 31: DTD-diagram for COMPANY-DOC-INFO

Child elements <company-ref> <doc-label> <private-codes> <entity-name>

parent elements <company-doc-infos>

Table 30: Attributes for COMPANY-DOC-INFO

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description This element takes company specific information about a document resp. a subpart of a document.

Example

### 3.16 COMPANY-DOC-INFOS



Figure 32: DTD-diagram for COMPANY-DOC-INFOS

Child elements <company-doc-info>

parent elements <admin-data>

Table 31: Attributes for COMPANY-DOC-INFOS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Company-specific information for administrative data

Example

### 3.17 COMPANY-REF



Figure 33: DTD-diagram for COMPANY-REF

Child elements none

parent elements <company-doc-info> <company-revision-info>

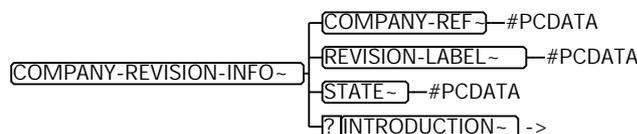
**Table 32: Attributes for COMPANY-REF**

Name	Type	Class	Value	Remark
[COMPANY]	idref	required		
[HYNAMES]	names	fixed	LINKEND COMPANY	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		

Description Reference to a company. The semantic of the reference is given by the context.

Example

### 3.18 COMPANY-REVISION-INFO



**Figure 34: DTD-diagram for COMPANY-REVISION-INFO**

Child elements **<company-ref>** **<revision-label>** **<state>** **<introduction>**

parent elements **<company-revision-infos>**

**Table 33: Attributes for COMPANY-REVISION-INFO**

Name	Type	Class	Value	Remark
[F-CHILD-TYPE]	cdata	fixed	state:selection	
[S]	cdata	implied		

Description Company-specific information concerning a certain revision of the document.

Example

### 3.19 COMPANY-REVISION-INFOS



**Figure 35: DTD-diagram for COMPANY-REVISION-INFOS**

Child elements **<company-revision-info>**

parent elements **<doc-revision>**

**Table 34: Attributes for COMPANY-REVISION-INFOS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Collection of all company specific revision information on an entity or a fragment.



Example

## 3.20 COND

`[COND~]~[P~] ->`

Figure 36: DTD-diagram for COND

Child elements `<p>`

parent elements `<prm-char>`

Table 35: Attributes for COND

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Condition for parameters. This condition is given as informal description.

Example



## 4 COUNT ... COUNT

### 4.1 COUNT

`<COUNT>`—#PCDATA

**Figure 37: DTD-diagram for COUNT**

Child elements none

parent elements `<sw-param-value-block>`

**Table 36: Attributes for COUNT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Number of elements for one parameter block.

Example

## 5 DATE ... DOC-REVISIONS

### 5.1 DATE

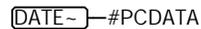


Figure 38: DTD-diagram for DATE

Child elements none

parent elements **<doc-revision>** **<schedule>**

Table 37: Attributes for DATE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Date information according to local rules resp. agreed standards. If the DTD is configured multilingual, the date can be specified in multiple languages.

Example

or as an international date

### 5.2 DATE-1



Figure 39: DTD-diagram for DATE-1

Child elements none

parent elements **<std>** **<xdoc>**

Table 38: Attributes for DATE-1

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Date information according to local rules resp. agreed standards. Even if the DTD is configured multilingual, the date **can not** be specified in multiple languages.

### 5.3 DEF

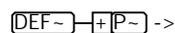


Figure 40: DTD-diagram for DEF

Child elements **<p>**

parent elements **<def-item>**

Table 39: Attributes for DEF

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 5.4 DEF-ITEM

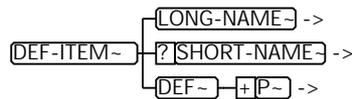


Figure 41: DTD-diagram for DEF-ITEM

Child elements `<long-name>` `<short-name>` `<def>`

parent elements `<def-list>`

Table 40: Attributes for DEF-ITEM

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	DEF-ITEM	
[HELP-ENTRY]	cdata	implied		
[ID]	id	required		
[S]	cdata	implied		

## 5.5 DEF-LIST

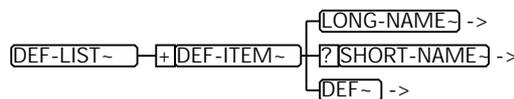


Figure 42: DTD-diagram for DEF-LIST

Child elements `<def-item>`

parent elements `<add-info>` `<chapter>` `<entry>` `<introduction>` `<item>` `<labeled-item>` `<ncoi-1>` `<remark>` `<sw-addressing-method-desc>` `<sw-application-notes>` `<sw-carb-doc>` `<sw-code-syntax-desc>` `<sw-function-desc>` `<sw-maintenance-notes>` `<sw-param-record-layout-desc>` `<sw-test-spec>` `<topic-1>` `<topic-2>`

Table 41: Attributes for DEF-LIST

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 5.6 DEMARCATION-OTHER-PROJECTS

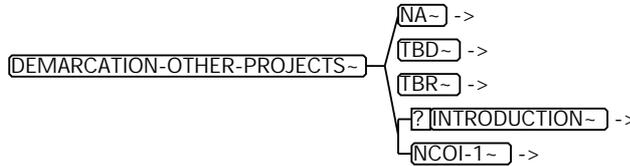


Figure 43: DTD-diagram for DEMARCATION-OTHER-PROJECTS

Child elements <na> <tbd> <tbr> <introduction> <ncoi-1>

parent elements <general-project-data>

Table 42: Attributes for DEMARCATION-OTHER-PROJECTS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Informal description of relationship to similar projects.

## 5.7 DEPARTMENT



Figure 44: DTD-diagram for DEPARTMENT

Child elements none

parent elements <team-member>

Table 43: Attributes for DEPARTMENT

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Department of the <team-member>

## 5.8 DESC

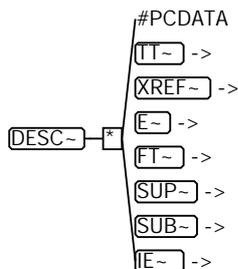


Figure 45: DTD-diagram for DESC

Child elements <tt> <xref> <e> <ft> <sup> <sub> <ie>

parent elements **<figure> <overall-project> <prm> <project> <sw-data-dictionary> <sw-param> <sw-param-content> <sw-param-contents> <sw-physic-type> <sw-physic-type-1> <sw-variable> <tbd>**

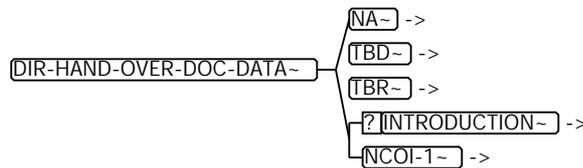
**Table 44: Attributes for DESC**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Informal, concise description of the subject in question. This is usually one or two sentences.

Example

## 5.9 DIR-HAND-OVER-DOC-DATA



**Figure 46: DTD-diagram for DIR-HAND-OVER-DOC-DATA**

Child elements **<na> <tbd> <tbr> <introduction> <ncoi-1>**

parent elements **<general-project-data>**

**Table 45: Attributes for DIR-HAND-OVER-DOC-DATA**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 5.10 DOC-LABEL

**DOC-LABEL~** -#PCDATA

**Figure 47: DTD-diagram for DOC-LABEL**

Child elements none

parent elements **<company-doc-info>**

**Table 46: Attributes for DOC-LABEL**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description This is the label of the document, serving as a company specific title or name of the document.

Example

## 5.11 DOC-REVISION

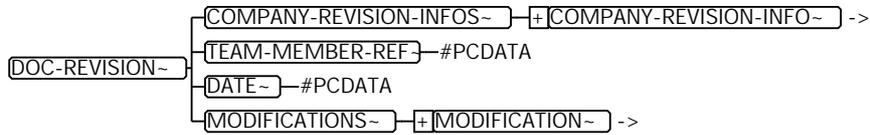


Figure 48: DTD-diagram for DOC-REVISION

Child elements **<company-revision-infos>** **<team-member-ref>** **<date>** **<modifications>**

parent elements **<doc-revisions>**

Table 47: Attributes for DOC-REVISION

Name	Type	Class	Value	Remark
[F-CHILD-TYPE]	cdata	fixed	date:date	
[S]	cdata	implied		

Description This element gets all information of the document revision in question.

Example

## 5.12 DOC-REVISIONS

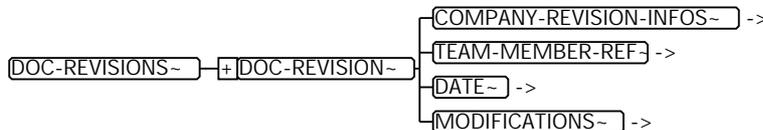


Figure 49: DTD-diagram for DOC-REVISIONS

Child elements **<doc-revision>**

parent elements **<admin-data>**

Table 48: Attributes for DOC-REVISIONS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description This is the collection of all known revisions of the document thus establishing the revision history. It is highly recommended that the most actual revision is on top.

## 6 E ... ENTRY

### 6.1 E

#PCDATA

Figure 50: DTD-diagram for E

Child elements none

parent elements <change> <desc> <indent-sample> <item-label> <p> <reason> <tbr>

Table 49: Attributes for E

Name	Type	Class	Value	Remark
[S]	cdata	implied		
[TYPE]	nmtkgrp	default	BOLD BOLD ITALIC	

### 6.2 EMAIL

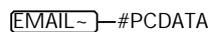
#PCDATA

Figure 51: DTD-diagram for EMAIL

Child elements none

parent elements <team-member>

Table 50: Attributes for EMAIL

Name	Type	Class	Value	Remark
[S]	cdata	implied		

### 6.3 ENTITY-NAME

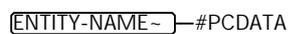
#PCDATA

Figure 52: DTD-diagram for ENTITY-NAME

Child elements none

parent elements <company-doc-info>

Table 51: Attributes for ENTITY-NAME

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Filename of the document fragment (if there is one). This is used to divide a document in multiple entities. *SGML tools* do not reflect this (physical) fragmentation on a logical level. Therefore, it must be entered manually here.

## 6.4 ENTRY

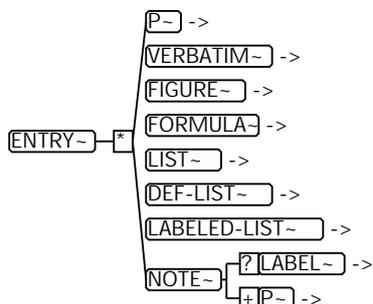


Figure 53: DTD-diagram for ENTRY

Child elements `<p>` `<verbatim>` `<figure>` `<formula>` `<list>` `<def-list>` `<labeled-list>` `<note>`

parent elements `<row>`

Table 52: Attributes for ENTRY

Name	Type	Class	Value	Remark
[ALIGN]	nmtkgrp	implied	LEFT RIGHT CENTER JUSTIFY CHAR	
[CHAR]	cdata	implied		
[CHAROFF]	nutoken	implied		
[COLNAME]	nmtoken	implied		
[COLSEP]	number	implied		
[MOREROWS]	number	default	0	
[NAMEEND]	nmtoken	implied		
[NAMEST]	nmtoken	implied		
[ROTATE]	number	default	0	
[ROWSEP]	number	implied		
[S]	cdata	implied		
[SPANNAME]	nmtoken	implied		
[VALIGN]	nmtkgrp	default	TOP TOP BOTTOM MIDDLE	

## 7 FAX ... FUNCTION-REF

### 7.1 FAX

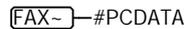


Figure 54: DTD-diagram for FAX

Child elements none

parent elements <team-member>

Table 53: Attributes for FAX

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

### 7.2 FIGURE

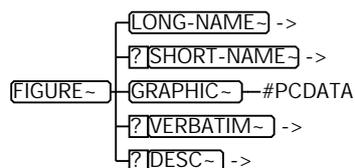


Figure 55: DTD-diagram for FIGURE

Child elements <long-name> <short-name> <graphic> <verbatim> <desc>

parent elements <add-info> <chapter> <entry> <introduction> <item> <labeled-item> <ncoi-1> <remark> <sw-addressing-method-desc> <sw-application-notes> <sw-carb-doc> <sw-code-syntax-desc> <sw-function-def> <sw-function-desc> <sw-maintenance-notes> <sw-param-record-layout-desc> <sw-test-spec> <topic-1> <topic-2>

Table 54: Attributes for FIGURE

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	FIGURE	
[FLOAT]	nmtkgrp	implied	FLOAT NO-FLOAT	
[HELP-ENTRY]	CDATA	implied		
[ID]	id	required		
[S]	CDATA	implied		

Description Graphics can be linked using this element.

Example

## 7.3 FILE

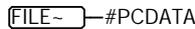


Figure 56: DTD-diagram for FILE

Child elements none

parent elements `<std>` `<xdoc>` `<xfile>`

Table 55: Attributes for FILE

Name	Type	Class	Value	Remark
[FILENAME]	cdata	required		
[NOTATION]	cdata	required		
[S]	cdata	implied		
[TOOL]	cdata	required		
[TOOL-VERSION]	cdata	required		

Description Information on a file name for referencing an external file (e.g. a standard). With respect to the limited capabilities of SGML's entity and notation mechanism this is not used here.

Example

## 7.4 FORMULA

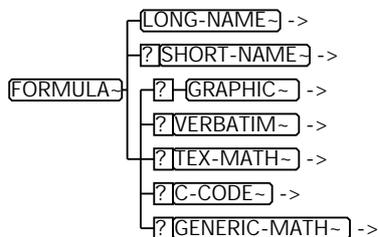


Figure 57: DTD-diagram for FORMULA

Child elements `<long-name>` `<short-name>` `<graphic>` `<verbatim>` `<tex-math>` `<c-code>` `<generic-math>`

parent elements `<add-info>` `<chapter>` `<entry>` `<introduction>` `<item>` `<labeled-item>` `<ncoi-1>` `<remark>` `<sw-addressing-method-desc>` `<sw-application-notes>` `<sw-carb-doc>` `<sw-code-syntax-desc>` `<sw-function-desc>` `<sw-maintenance-notes>` `<sw-param-record-layout-desc>` `<sw-test-spec>` `<topic-1>` `<topic-2>`

Table 56: Attributes for FORMULA

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	FORMULA	
[ID]	id	required		
[S]	cdata	implied		

## 7.5 FT

—#PCDATA

**Figure 58: DTD-diagram for FT**

Child elements none

parent elements **<change>** **<desc>** **<indent-sample>** **<item-label>** **<p>** **<reason>** **<tbr>**

**Table 57: Attributes for FT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 7.6 FUNCTION-REF

—#PCDATA

**Figure 59: DTD-diagram for FUNCTION-REF**

Child elements none

parent elements **<sw-fulfils>**

**Table 58: Attributes for FUNCTION-REF**

Name	Type	Class	Value	Remark
[FUNCTION]	idref	required		
[HYNAMES]	names	fixed	LINKEND FUNCTION	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		

Description This is an external reference to a **<function>** within **<behaviour>** in an instance of *MSRSYS DTD*.

Example

## 8 GENERAL-PROJECT-DATA ... GRAPHIC

### 8.1 GENERAL-PROJECT-DATA

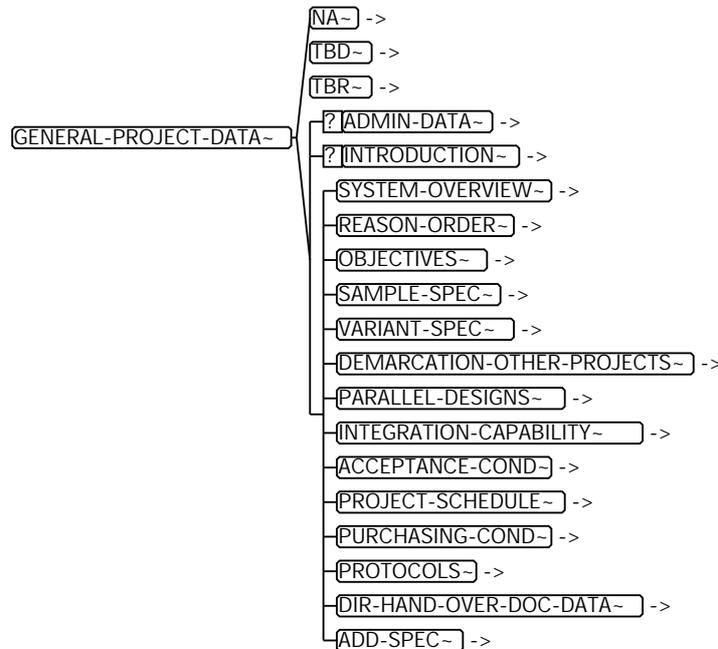


Figure 60: DTD-diagram for GENERAL-PROJECT-DATA

Child elements `<na>` `<tdb>` `<tbr>` `<admin-data>` `<introduction>` `<system-overview>` `<reason-order>` `<objectives>` `<sample-spec>` `<variant-spec>` `<demarcation-other-projects>` `<parallel-designs>` `<integration-capability>` `<acceptance-cond>` `<project-schedule>` `<purchasing-cond>` `<protocols>` `<dir-hand-over-doc-data>` `<add-spec>`

parent elements `<company>`

Table 59: Attributes for GENERAL-PROJECT-DATA

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description General project data.

Example

### 8.2 GENERIC-MATH

`<GENERIC-MATH~>#PCDATA`

Figure 61: DTD-diagram for GENERIC-MATH

Child elements none

parent elements `<formula>`



Table 60: Attributes for GENERIC-MATH

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description This element is intended for the definition of semantic math description which can be processed by math processors. Actually there is no recommendation for the language of the formula specification or usage of a special rendering system.

Example

## 8.3 GRAPHIC

`<GRAPHIC~>`—#PCDATA

Figure 62: DTD-diagram for GRAPHIC

Child elements none

parent elements `<figure>` `<formula>`

Table 61: Attributes for GRAPHIC

Name	Type	Class	Value	Remark
[CATEGORY]	nmtkgrp	implied	BARCODE CONCEPTUAL ENGINEERING FLOWCHART GRAPH LOGO SCHEMATIC WAVEFORM	
[FILENAME]	cdata	required		
[FIT]	number	default	0	
[HEIGHT]	cdata	implied		
[NOTATION]	cdata	required		
[S]	cdata	implied		
[SCALE]	cdata	implied		
[WIDTH]	cdata	implied		

Description The treatment of the graphic is determined by the attributes of `<graphic>`:

Do not enter annotating text to `<long-name>`in`<figure>` or `<table>`(like *Figure 1: ...*). This embellishment is the task of the processing system, not of the author. If the author adds these things, they will be there twice since the *rendition system* will add it again.

**[category]** Denotes the category of the graphic. This information can be used to generate more specific list of figures

**[filename]** Denotes the system filename where the *rendition system* can find the graphic. This is not necessarily the final format. It is up to the *rendition system* to locate the graphic in the company specific environment, to change the file extension to get the appropriate graphic representation.

The type of this attribute can be turned from *SDATA* to *ENTITY* in the DTD file in order to allow *SGML tools* access to the file using its *entity*

*manager*. In this case, the entity name should be chosen in the style of a filename (e.g. *crpctmt.wmf*)<sup>1</sup>.

- [fit]** 0 figure is placed in original size. If it does not fit on the page or the available space, it is scaled down.
- 1 the figure is scaled up or down to fit the page as possible. This value will be ignored if **[width]** or **[height]** is specified in addition.
- 2 the figure is rotated counterclockwise by 90° if it is landscape and is wider than the actual text area. It is scaled down to the page size if it does not fit otherwise. This value will be ignored if **[width]** or **[height]** is specified in addition.
- 3 the figure is always rotated counterclockwise by 90°. If it does not fit on the page it will be scaled down. If **[width]** or **[height]** is specified in addition, the figure will be rotated and then scaled to the specified values.
- 4 the figure is always rotated counterclockwise by 90° and scaled up or down for best fit on the page. This value will be ignored if **[width]** or **[height]** is specified in addition.
- [height]** If this attribute has a value, the figure will be scaled to the defined height which is a real value with dimensions (e.g. "10cm", "150mm", "12.5in"). If also **[width]** is specified the figure will be distorted. This value always specifies the width of the "figure box" on the page after possible scaling/rotating.
- [notation]** This attribute specifies the format of the graphic file if used by an *SGML Application* supporting notations.
- [scale]** If this attribute receives a value, the figure will be scaled by the given factor which must be a signed real number. Numbers greater 1 increase the size of the figure, values less than 1 make the figure smaller. For example with *scale="0.5"* the a figure of the size 10x10 cm will appear as 5\*5cm.
- [width]** If this attribute has a value, the figure will be scaled to the defined width which is a real value with dimensions (e.g. "10cm", "150mm", "12.5in"). If also **[height]** is specified the figure will be distorted. This value always specifies the width of the "figure box" on the page after possible scaling/rotating.

The scaling attribute precedence is:

- **[scale]** has precedence over all
- **[fit]** has precedence over **[width]** and/or **[height]**

Example



## 9 HOMEPAGE ... HOMEPAGE

### 9.1 HOMEPAGE

`{HOMEPAGE}`—#PCDATA

**Figure 63: DTD-diagram for HOMEPAGE**

Child elements none

parent elements `<team-member>`

**Table 62: Attributes for HOMEPAGE**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 10 IE ... ITEM-LABEL

### 10.1 IE

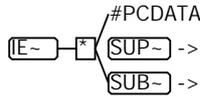


Figure 64: DTD-diagram for IE

Child elements `<sup>` `<sub>`

parent elements `<change>` `<desc>` `<indent-sample>` `<item-label>` `<long-name>` `<long-name-1>` `<p>` `<reason>` `<tr>`

Table 63: Attributes for IE

Name	Type	Class	Value	Remark
[S]	CDATA	implied		
[TYPE]	CDATA	implied		

Description This is an index entry. The contents only appears in the index, not at the location where it is defined. This allows to define slightly different wording for the index than in the text itself.

Example

### 10.2 INDENT-SAMPLE

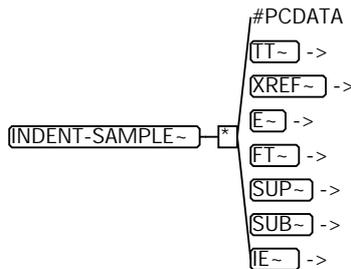


Figure 65: DTD-diagram for INDENT-SAMPLE

Child elements `<tt>` `<xref>` `<e>` `<ft>` `<sup>` `<sub>` `<ie>`

parent elements `<labeled-list>`

Table 64: Attributes for INDENT-SAMPLE

Name	Type	Class	Value	Remark
[ITEM-LABEL-POS]	nmtkgrp	default	NO-NEWLINE NO-NEWLINE NEW- LINE NEWLINE-IF- NECESSARY	
[S]	CDATA	implied		

## 10.3 INTEGRATION-CAPABILITY

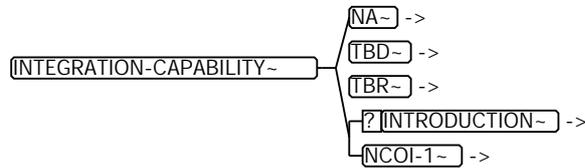


Figure 66: DTD-diagram for INTEGRATION-CAPABILITY

Child elements `<na>` `<tbd>` `<tbr>` `<introduction>` `<ncoi-1>`

parent elements `<general-project-data>`

Table 65: Attributes for INTEGRATION-CAPABILITY

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 10.4 INTRODUCTION

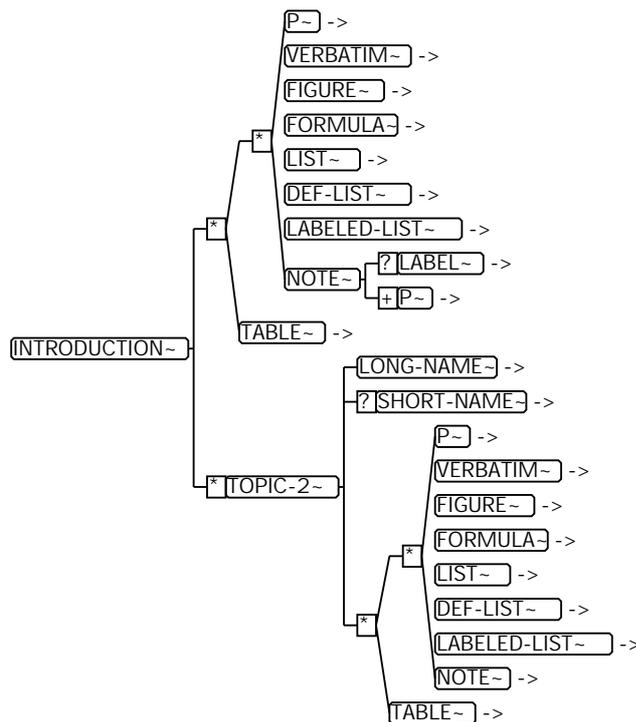


Figure 67: DTD-diagram for INTRODUCTION

Child elements `<p>` `<verbatim>` `<figure>` `<formula>` `<list>` `<def-list>` `<labeled-list>` `<note>` `<table>` `<topic-2>`

parent elements `<acceptance-cond>` `<add-spec>` `<company-revision-info>` `<demarcation-other-projects>` `<dir-hand-over-doc-data>` `<general-project-data>` `<integration-capability>` `<msrsw>` `<objectives>` `<parallel-designs>` `<project-schedule>` `<protocols>` `<purchasing-cond>` `<reason-order>` `<sample-spec>` `<sw-`

**<data-dictionary-spec> <sw-function-spec> <sw-glossary> <sw-param-contents-spec> <system-overview> <variant-spec>**

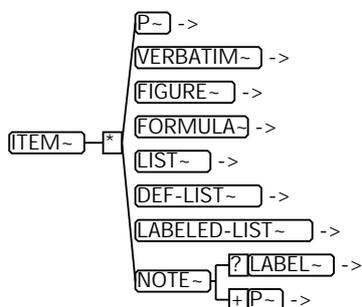
**Table 66: Attributes for INTRODUCTION**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description This is used to give a short introduction about the subject in question. It exists at the beginning of its parent element. It is not intended to use **<introduction>** for a full fledged description.

Example

## 10.5 ITEM



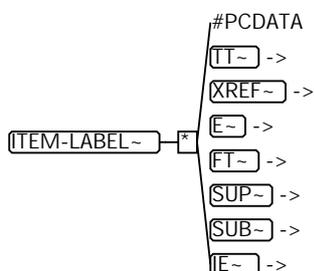
**Figure 68: DTD-diagram for ITEM**

Child elements **<p> <verbatim> <figure> <formula> <list> <def-list> <labeled-list> <note>**  
parent elements **<list>**

**Table 67: Attributes for ITEM**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 10.6 ITEM-LABEL



**Figure 69: DTD-diagram for ITEM-LABEL**

Child elements **<tt> <xref> <e> <ft> <sup> <sub> <ie>**  
parent elements **<labeled-item>**



**Table 68: Attributes for ITEM-LABEL**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

# 11 LABEL ... LONG-NAME-1

## 11.1 LABEL

`[LABEL~]`—#PCDATA

Figure 70: DTD-diagram for LABEL

Child elements none

parent elements `<annotation>` `<labels>` `<note>` `<overall-project>` `<prms>` `<project>`

Table 69: Attributes for LABEL

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description Serves as a title (similar to `<long-name>`) for an object that must not, and cannot, be referenced, i.e. possesses no `<short-name>` and no `[id]`.

A `<label>` within `<sw-param-value-block>` is a long designation for the characteristic values of a characteristic values block.

Example

## 11.2 LABELED-ITEM

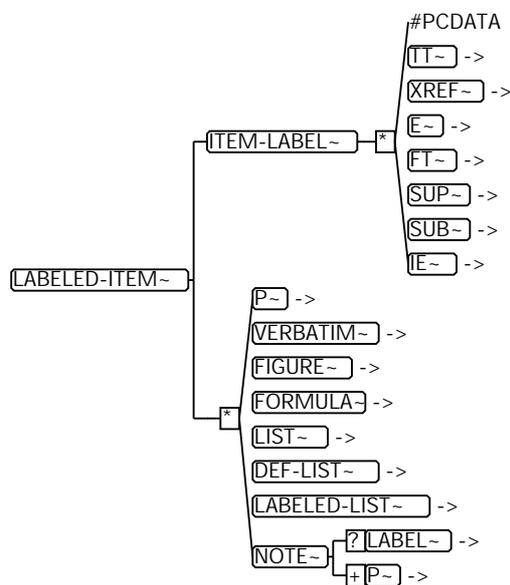


Figure 71: DTD-diagram for LABELED-ITEM

Child elements `<item-label>` `<p>` `<verbatim>` `<figure>` `<formula>` `<list>` `<def-list>` `<labeled-list>` `<note>`

parent elements `<labeled-list>`

Table 70: Attributes for LABELED-ITEM

Name	Type	Class	Value	Remark
[HELP-ENTRY]	cdata	implied		
[S]	cdata	implied		

## 11.3 LABELED-LIST

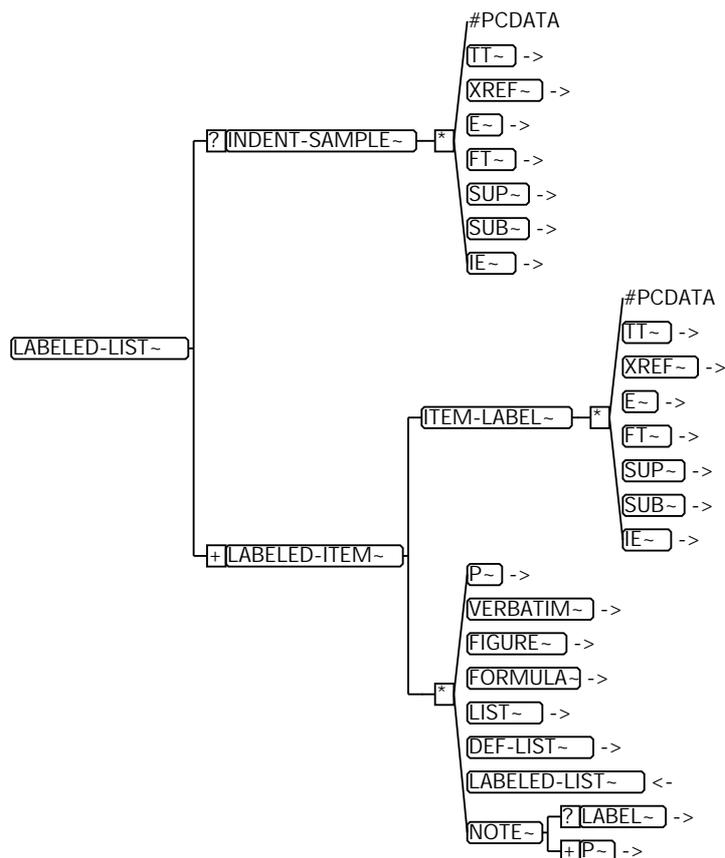


Figure 72: DTD-diagram for LABELED-LIST

Child elements `<indent-sample>` `<labeled-item>`

parent elements `<add-info>` `<chapter>` `<entry>` `<introduction>` `<item>` `<labeled-item>` `<ncoi-1>` `<remark>` `<sw-addressing-method-desc>` `<sw-application-notes>` `<sw-carb-doc>` `<sw-code-syntax-desc>` `<sw-function-desc>` `<sw-maintenance-notes>` `<sw-param-record-layout-desc>` `<sw-test-spec>` `<topic-1>` `<topic-2>`

Table 71: Attributes for LABELED-LIST

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 11.4 LABELS

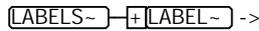
 ->

Figure 73: DTD-diagram for LABELS

Child elements <label>

parent elements <sw-param-axis-values>

Table 72: Attributes for LABELS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Set of labels for axis values giving the meaning of each individual value in an <sw-param-value-block>.

Example

## 11.5 LANGUAGE

 -#PCDATA

Figure 74: DTD-diagram for LANGUAGE

Child elements none

parent elements <admin-data>

Table 73: Attributes for LANGUAGE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Identifies the masterlanguage for <admin-data>

Example

## 11.6 LIST

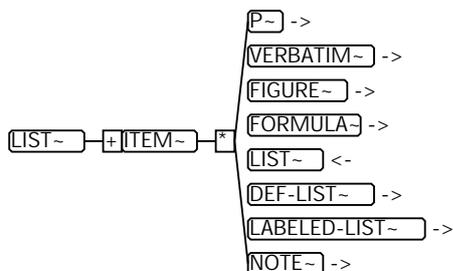
 ->

Figure 75: DTD-diagram for LIST

Child elements <item>

parent elements **<add-info> <chapter> <entry> <introduction> <item> <labeled-item> <ncoi-1> <remark> <sw-addressing-method-desc> <sw-application-notes> <sw-carb-doc> <sw-code-syntax-desc> <sw-function-desc> <sw-maintenance-notes> <sw-param-record-layout-desc> <sw-test-spec> <topic-1> <topic-2>**

**Table 74: Attributes for LIST**

Name	Type	Class	Value	Remark
[S]	cdata	implied		
[TYPE]	nmtkgrp	required	UNNUMBER NUMBER	

## 11.7 LOCS



**Figure 76: DTD-diagram for LOCS**

Child elements **<nameloc>**

parent elements **<mrsrw>**

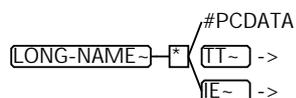
**Table 75: Attributes for LOCS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description This element is used for cross document/entity referencing (HyTime referencing using namelocs)

Example

## 11.8 LONG-NAME



**Figure 77: DTD-diagram for LONG-NAME**

Child elements **<tt> <ie>**

parent elements **<chapter> <company> <def-item> <figure> <formula> <nameloc> <prm> <sample> <sw-addressing-method> <sw-code-syntax> <sw-compu-method> <sw-function> <sw-param> <sw-param-record-layout> <sw-physic-type> <sw-unit> <sw-variable> <table> <team-member> <topic-1> <topic-2> <variant-char> <variant-def>**

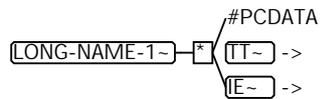
**Table 76: Attributes for LONG-NAME**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Long designation, e.g. "Engine temperature".

Example

## 11.9 LONG-NAME-1



**Figure 78: DTD-diagram for LONG-NAME-1**

Child elements `<tt>` `<ie>`

parent elements `<std>` `<xdoc>` `<xfile>`

**Table 77: Attributes for LONG-NAME-1**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 12 MAX ... MSRSW

### 12.1 MAX

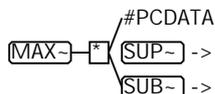


Figure 79: DTD-diagram for MAX

Child elements **<sup>** **<sub>**

parent elements **<prm-char>**

Table 78: Attributes for MAX

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Maximum value of a parameter

### 12.2 MAX-COUNT



Figure 80: DTD-diagram for MAX-COUNT

Child elements none

parent elements **<sw-axis-individual>** **<sw-axis-shift-offset>**

Table 79: Attributes for MAX-COUNT

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Maximum number of datapoints

Example

### 12.3 MIN

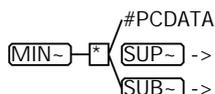


Figure 81: DTD-diagram for MIN

Child elements **<sup>** **<sub>**

parent elements **<prm-char>**

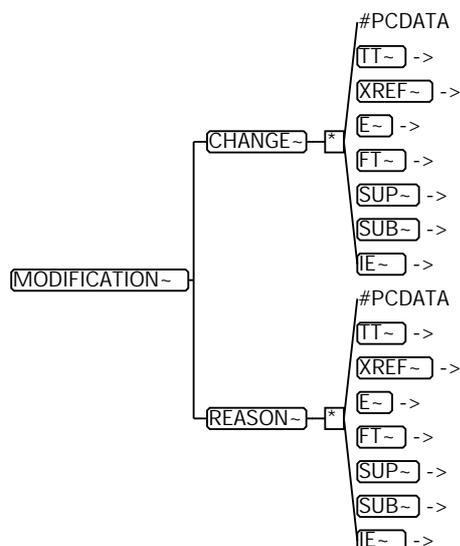
**Table 80: Attributes for MIN**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Minimum value for prm-char

Example

## 12.4 MODIFICATION



**Figure 82: DTD-diagram for MODIFICATION**

Child elements <change> <reason>

parent elements <modifications>

**Table 81: Attributes for MODIFICATION**

Name	Type	Class	Value	Remark
[S]	cdata	implied		
[TYPE]	nmtkgrp	required	CONTENT-RELATED DOC-RELATED	

Description

Example

## 12.5 MODIFICATIONS



**Figure 83: DTD-diagram for MODIFICATIONS**

Child elements <modification>



parent elements **<doc-revision>**

**Table 82: Attributes for MODIFICATIONS**

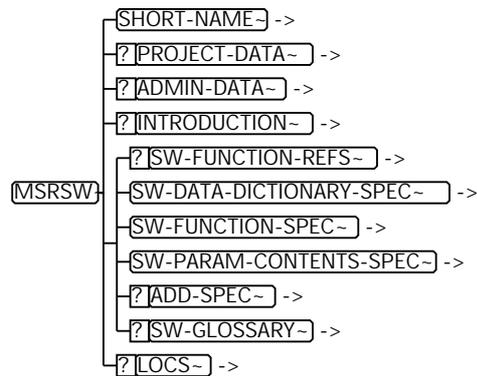
Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description

Example

## 12.6

## MSRSW



**Figure 84: DTD-diagram for MSRSW**

Child elements **<short-name> <project-data> <admin-data> <introduction> <sw-function-refs> <sw-data-dictionary-spec> <sw-function-spec> <sw-param-contents-spec> <add-spec> <sw-glossary> <locs>**

parent elements none

**Table 83: Attributes for MSRSW**

Name	Type	Class	Value	Remark
[F-NAMESPACE]	names	fixed	COMPANY SAMPLE TEAM-MEMBER VARIANT-CHAR VARIANT-DEF XDOC XFILE STD DEF-ITEM FIGURE FORMULA TABLE PRM TOPIC CHAPTER SW-ADDRESSING-METHOD SW-CODE-SYNTAX SW-COMPU-METHOD SW-FUNCTION SW-PARAM SW-PARAM-RECORD-LAYOUT SW-PHYSIC-TYPE SW-UNIT SW-VARIABLE	



**Table 83 (Cont.): Attributes for MSRSW**

Name	Type	Class	Value	Remark
[F-PUBID]	cdata	fixed	-//MSR//DTD MSR SOFTWARE DT-D:V1.1.0:MSRSW.DTD//EN	
[HYTIME]	name	fixed	HYDOC	
[PUBID]	cdata	default	-//MSR//DTD MSR SOFTWARE DT-D:V1.1.0:MSRSW.DTD//EN	
[S]	cdata	implied		

Description Root element

Example

## 13 NA ... NUMBER

### 13.1 NA

#PCDATA

Figure 85: DTD-diagram for NA

Child elements none

parent elements <acceptance-cond> <add-spec> <demarcation-other-projects> <dir-hand-over-doc-data> <general-project-data> <integration-capability> <objectives> <parallel-designs> <project-schedule> <protocols> <purchasing-cond> <reason-order> <sample-spec> <sw-data-dictionary-spec> <sw-function-spec> <sw-glossary> <sw-param-contents-spec> <system-overview> <variant-spec>

Table 84: Attributes for NA

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description This element is used instead of sub-structures if certain statements are not relevant ("not applicable").

Example

### 13.2 NAMELOC



- LONG-NAME ->
- SHORT-NAME ->
- NMLIST -#PCDATA

Figure 86: DTD-diagram for NAMELOC

Child elements <long-name> <short-name> <nmlist>

parent elements <locs>

Table 85: Attributes for NAMELOC

Name	Type	Class	Value	Remark
[HYTIME]	name	fixed	NAMELOC	
[ID]	id	required		
[S]	CDATA	implied		

### 13.3 NCOI-1

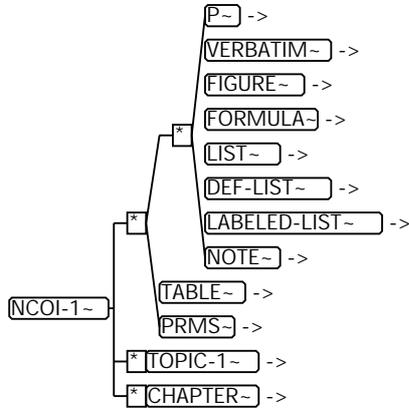


Figure 87: DTD-diagram for NCOI-1

Child elements `<p>` `<verbatim>` `<figure>` `<formula>` `<list>` `<def-list>` `<labeled-list>` `<note>` `<table>` `<prms>` `<topic-1>` `<chapter>`

parent elements `<acceptance-cond>` `<add-spec>` `<demarcation-other-projects>` `<dir-hand-over-doc-data>` `<integration-capability>` `<objectives>` `<parallel-designs>` `<project-schedule>` `<protocols>` `<purchasing-cond>` `<reason-order>` `<sample>` `<sw-function-def>` `<sw-glossary>` `<system-overview>`

Table 86: Attributes for NCOI-1

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description This is a general element that contains informal and non-software-specific structures ("none coded information").

Example

### 13.4 NMLIST

`<NMLIST~>` #PCDATA

Figure 88: DTD-diagram for NMLIST

Child elements none

parent elements `<nameloc>`

Table 87: Attributes for NMLIST

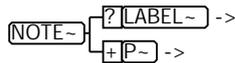
Name	Type	Class	Value	Remark
[DOCORSUB]	entity	implied		
[HYTIME]	name	fixed	NMLIST	
[NAMETYPE]	nmtgrp	default	ELEMENT ENTITY ELEMENT	



**Table 87 (Cont.): Attributes for NMLIST**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 13.5 NOTE



**Figure 89: DTD-diagram for NOTE**

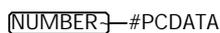
Child elements **<label>** **<p>**

parent elements **<add-info>** **<chapter>** **<entry>** **<introduction>** **<item>** **<labeled-item>** **<ncoi-1>** **<remark>** **<sw-addressing-method-desc>** **<sw-application-notes>** **<sw-carb-doc>** **<sw-code-syntax-desc>** **<sw-function-desc>** **<sw-maintenance-notes>** **<sw-param-record-layout-desc>** **<sw-test-spec>** **<topic-1>** **<topic-2>**

**Table 88: Attributes for NOTE**

Name	Type	Class	Value	Remark
[NOTE-TYPE]	nmtkgrp	required	CAUTION HINT TIP INSTRUCTION EX- ERCISE OTHER	
[S]	cdata	implied		
[USER-DEFINED-TYPE]	cdata	implied		

## 13.6 NUMBER



**Figure 90: DTD-diagram for NUMBER**

Child elements none

parent elements **<xdoc>**

**Table 89: Attributes for NUMBER**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 14 OBJECTIVES ... OVERALL-PROJECT

### 14.1 OBJECTIVES

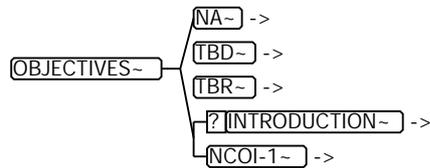


Figure 91: DTD-diagram for OBJECTIVES

Child elements `<na>` `<tbd>` `<tbr>` `<introduction>` `<ncoi-1>`

parent elements `<general-project-data>`

Table 90: Attributes for OBJECTIVES

Name	Type	Class	Value	Remark
[S]	cdata	implied		

### 14.2 OFFSET

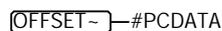


Figure 92: DTD-diagram for OFFSET

Child elements none

parent elements `<sw-axis-shift-offset>`

Table 91: Attributes for OFFSET

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description The datapoints are defined by an algorithm for fixed characteristics and maps.  
Example for an algorithm:  $\text{Datapoint}[i] = ( \dots \text{shift} ) * x + \text{offset}$

Example

### 14.3 OVERALL-PROJECT



Figure 93: DTD-diagram for OVERALL-PROJECT

Child elements `<label>` `<desc>`

parent elements `<project-data>`



**Table 92: Attributes for OVERALL-PROJECT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

# 15 P ... PURCHASING-COND

## 15.1 P

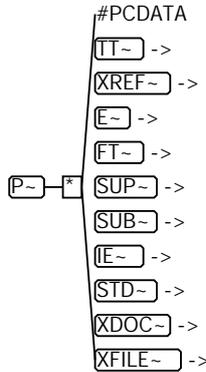


Figure 94: DTD-diagram for P

Child elements `<tt>` `<xref>` `<e>` `<ft>` `<sup>` `<sub>` `<ie>` `<std>` `<xdoc>` `<xfile>`

parent elements `<add-info>` `<annotation-text>` `<chapter>` `<cond>` `<def>` `<entry>` `<introduction>` `<item>` `<labeled-item>` `<ncoi-1>` `<note>` `<remark>` `<sw-addressing-method-desc>` `<sw-application-notes>` `<sw-carb-doc>` `<sw-code-syntax-desc>` `<sw-function-desc>` `<sw-maintenance-notes>` `<sw-param-record-layout-desc>` `<sw-test-spec>` `<topic-1>` `<topic-2>`

Table 93: Attributes for P

Name	Type	Class	Value	Remark
[HELP-ENTRY]	CDATA	implied		
[S]	CDATA	implied		

Description Paragraph. A paragraph can comprise text and the elements tt (technical text), xref (cross reference), e (text attribute like bold), ft (footnote), sup (superscript), sub (subscript), ie (index entry), std (standard), xdoc (external document), xfile (external file) in any arbitrary, though not hierarchical, order.

Example

## 15.2 PARALLEL-DESIGNS

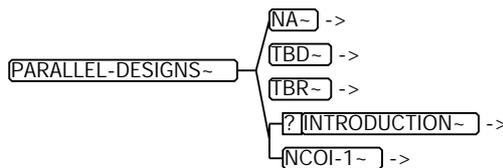


Figure 95: DTD-diagram for PARALLEL-DESIGNS

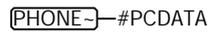
Child elements `<na>` `<tbd>` `<tbr>` `<introduction>` `<ncoi-1>`

parent elements `<general-project-data>`

**Table 94: Attributes for PARALLEL-DESIGNS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 15.3 PHONE

#PCDATA

**Figure 96: DTD-diagram for PHONE**

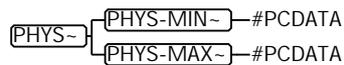
Child elements none

parent elements <team-member>

**Table 95: Attributes for PHONE**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 15.4 PHYS

#PCDATA

**Figure 97: DTD-diagram for PHYS**

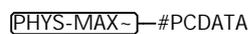
Child elements <phys-min> <phys-max>

parent elements <sw-limits> <sw-limits>

**Table 96: Attributes for PHYS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 15.5 PHYS-MAX

#PCDATA

**Figure 98: DTD-diagram for PHYS-MAX**

Child elements none

parent elements <phys>

**Table 97: Attributes for PHYS-MAX**

Name	Type	Class	Value	Remark
[S]	cdata	implied		



## 15.6 PHYS-MIN

`[PHYS-MIN~]`—#PCDATA

Figure 99: DTD-diagram for PHYS-MIN

Child elements none

parent elements `<phys>`

Table 98: Attributes for PHYS-MIN

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 15.7 POSITION

`[POSITION~]`—#PCDATA

Figure 100: DTD-diagram for POSITION

Child elements none

parent elements `<std>` `<xdoc>`

Table 99: Attributes for POSITION

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 15.8 PRIVATE-CODE

`[PRIVATE-CODE~]`—#PCDATA

Figure 101: DTD-diagram for PRIVATE-CODE

Child elements none

parent elements `<private-codes>`

Table 100: Attributes for PRIVATE-CODE

Name	Type	Class	Value	Remark
[S]	cdata	implied		
[TYPE]	cdata	implied		

Description

Example

## 15.9 PRIVATE-CODES

`[PRIVATE-CODES~]` `+ [PRIVATE-CODE~]` `#PCDATA`

Figure 102: DTD-diagram for PRIVATE-CODES

Child elements `<private-code>`

parent elements `<company-doc-info>`

Table 101: Attributes for PRIVATE-CODES

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description

Example

## 15.10 PRM

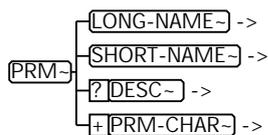


Figure 103: DTD-diagram for PRM

Child elements `<long-name>` `<short-name>` `<desc>` `<prm-char>`

parent elements `<prms>`

Table 102: Attributes for PRM

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	PRM	
[ID]	id	required		
[S]	cdata	implied		

Description A parameter model has been implemented in MSR. There are 2 possibilities given. A parameter can be specified either by defining `<abs>`, `<tol>` or by defining `<min>`, `<typ>` and `<max>`.

Example

## 15.11 PRM-CHAR

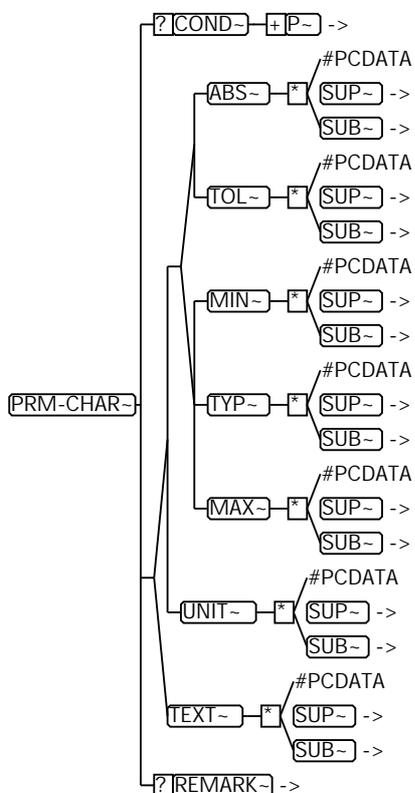


Figure 104: DTD-diagram for PRM-CHAR

Child elements <cond> <abs> <tol> <min> <typ> <max> <unit> <text> <remark>

parent elements <prm>

Table 103: Attributes for PRM-CHAR

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Description of the parameter characteristics of a parameter

Example

## 15.12 PRMS

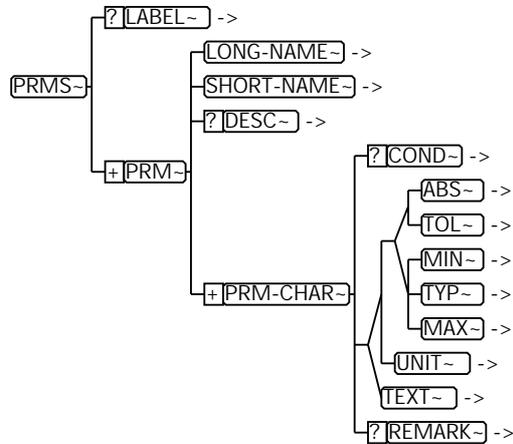


Figure 105: DTD-diagram for PRMS

Child elements <label> <prm>

parent elements <add-info> <chapter> <ncoi-1> <sw-addressing-method-desc> <sw-application-notes> <sw-carb-doc> <sw-code-syntax-desc> <sw-function-desc> <sw-maintenance-notes> <sw-param-record-layout-desc> <sw-test-spec> <topic-1>

Table 104: Attributes for PRMS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description List of parameters

Example

## 15.13 PROG-CODE

`PROG-CODE~`—#PCDATA

Figure 106: DTD-diagram for PROG-CODE

Child elements none

parent elements <sw-compu-method>

Table 105: Attributes for PROG-CODE

Name	Type	Class	Value	Remark
[LANG-SUBSET]	names	implied		
[PROG-LANG]	cdata	implied		
[S]	cdata	implied		
[USED-LIBS]	nmtokens	implied		

Description Description of conversion formulae in a programming language notation

Example

## 15.14 PROJECT



Figure 107: DTD-diagram for PROJECT

Child elements `<label>` `<desc>` `<companies>`

parent elements `<project-data>`

Table 106: Attributes for PROJECT

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Information regarding a project.

Example

## 15.15 PROJECT-DATA

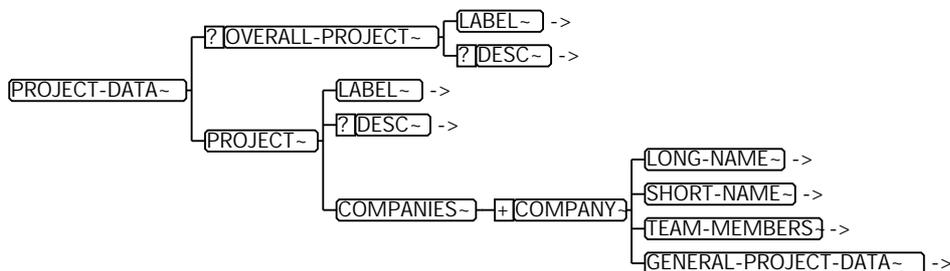


Figure 108: DTD-diagram for PROJECT-DATA

Child elements `<overall-project>` `<project>`

parent elements `<msrsw>`

Table 107: Attributes for PROJECT-DATA

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Project data

Example

## 15.16 PROJECT-SCHEDULE

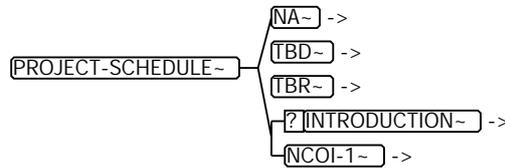


Figure 109: DTD-diagram for PROJECT-SCHEDULE

Child elements `<na>` `<tbd>` `<tbr>` `<introduction>` `<ncoi-1>`

parent elements `<general-project-data>`

Table 108: Attributes for PROJECT-SCHEDULE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 15.17 PROTOCOLS

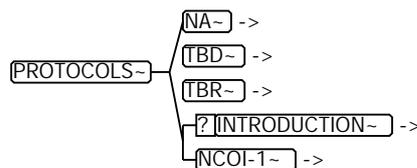


Figure 110: DTD-diagram for PROTOCOLS

Child elements `<na>` `<tbd>` `<tbr>` `<introduction>` `<ncoi-1>`

parent elements `<general-project-data>`

Table 109: Attributes for PROTOCOLS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 15.18 PUBLISHER



Figure 111: DTD-diagram for PUBLISHER

Child elements none

parent elements `<xdoc>`

Table 110: Attributes for PUBLISHER

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 15.19 PURCHASING-COND

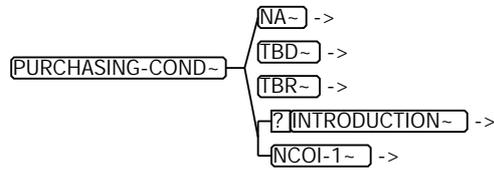


Figure 112: DTD-diagram for PURCHASING-COND

Child elements <na> <tbd> <tbr> <introduction> <ncoi-1>

parent elements <general-project-data>

Table 111: Attributes for PURCHASING-COND

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 16 REASON ... ROW

### 16.1 REASON

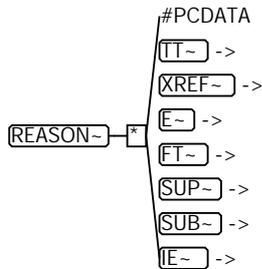


Figure 113: DTD-diagram for REASON

Child elements `<tt>` `<xref>` `<e>` `<ft>` `<sup>` `<sub>` `<ie>`

parent elements `<modification>`

Table 112: Attributes for REASON

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description

Example

### 16.2 REASON-ORDER

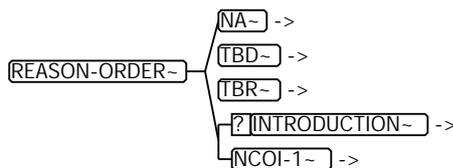


Figure 114: DTD-diagram for REASON-ORDER

Child elements `<na>` `<tbd>` `<tbr>` `<introduction>` `<ncoi-1>`

parent elements `<general-project-data>`

Table 113: Attributes for REASON-ORDER

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 16.3 REMARK

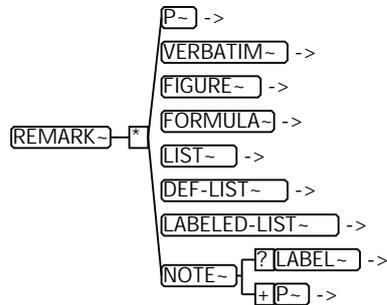


Figure 115: DTD-diagram for REMARK

Child elements `<p>` `<verbatim>` `<figure>` `<formula>` `<list>` `<def-list>` `<labeled-list>` `<note>`

parent elements `<prm-char>` `<sw-physic-type>` `<sw-physic-type-1>`

Table 114: Attributes for REMARK

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 16.4 REVISION-LABEL

`REVISION-LABEL~`—#PCDATA

Figure 116: DTD-diagram for REVISION-LABEL

Child elements none

parent elements `<company-revision-info>`

Table 115: Attributes for REVISION-LABEL

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Revision

Example

## 16.5 ROLE

`ROLE~`—#PCDATA

Figure 117: DTD-diagram for ROLE

Child elements none

parent elements `<roles>`

Table 116: Attributes for ROLE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 16.6 ROLES

$\boxed{\text{ROLES-}}$   $\boxed{+}$   $\boxed{\text{ROLE-}}$   $\boxed{-}$  #PCDATA

Figure 118: DTD-diagram for ROLES

Child elements **<role>**

parent elements **<team-member>**

Table 117: Attributes for ROLES

Name	Type	Class	Value	Remark
[F-CHILD-TYPE]	cdata	fixed	role:selection	
[S]	cdata	implied		

## 16.7 ROW

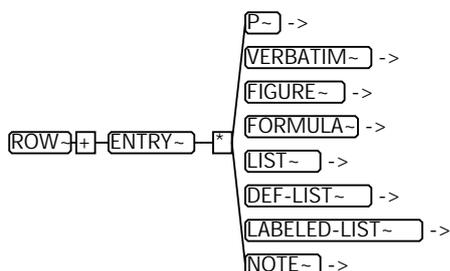


Figure 119: DTD-diagram for ROW

Child elements **<entry>**

parent elements **<tbody>** **<tfoot>** **<thead>**

Table 118: Attributes for ROW

Name	Type	Class	Value	Remark
[ROWSEP]	number	implied		
[S]	cdata	implied		

## 17 SAMPLE ... SW-ADDRESSING-METHODS

### 17.1 SAMPLE

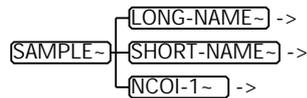


Figure 120: DTD-diagram for SAMPLE

Child elements <long-name> <short-name> <ncoi-1>

parent elements <samples>

Table 119: Attributes for SAMPLE

Name	Type	Class	Value	Remark
[F-CHILD-TYPE]	cdata	fixed	long-name:selection	
[F-ID-CLASS]	name	fixed	SAMPLE	
[ID]	id	required		
[S]	cdata	implied		

### 17.2 SAMPLE-REF

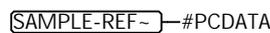


Figure 121: DTD-diagram for SAMPLE-REF

Child elements none

parent elements <schedule>

Table 120: Attributes for SAMPLE-REF

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND SAMPLE	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		
[SAMPLE]	idref	required		

### 17.3 SAMPLE-SPEC

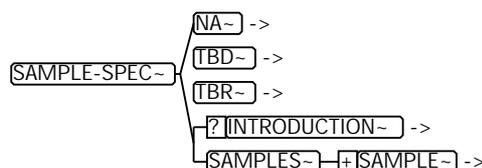


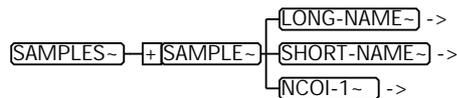
Figure 122: DTD-diagram for SAMPLE-SPEC

Child elements **<na>** **<tbd>** **<tbr>** **<introduction>** **<samples>**  
parent elements **<general-project-data>**

**Table 121: Attributes for SAMPLE-SPEC**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 17.4 SAMPLES



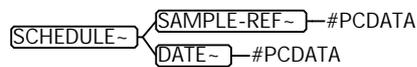
**Figure 123: DTD-diagram for SAMPLES**

Child elements **<sample>**  
parent elements **<sample-spec>**

**Table 122: Attributes for SAMPLES**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 17.5 SCHEDULE



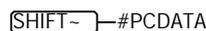
**Figure 124: DTD-diagram for SCHEDULE**

Child elements **<sample-ref>** **<date>**  
parent elements **<tbd>**

**Table 123: Attributes for SCHEDULE**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 17.6 SHIFT



**Figure 125: DTD-diagram for SHIFT**

Child elements none  
parent elements **<sw-axis-shift-offset>**

**Table 124: Attributes for SHIFT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description The datapoints are defined by an algorithm for fixed characteristics and fixed maps.  
Example for an algorithm:  $\text{Datapoint}[i] = ( \dots \text{shift} ) * x + \text{offset}$

Example

## 17.7 SHORT-NAME

`[SHORT-NAME]`-cdata

**Figure 126: DTD-diagram for SHORT-NAME**

Child elements none

parent elements `<chapter>` `<company>` `<def-item>` `<figure>` `<formula>` `<msrsw>` `<name-loc>` `<prm>` `<sample>` `<std>` `<sw-addressing-method>` `<sw-code-syntax>` `<sw-compu-method>` `<sw-function>` `<sw-param>` `<sw-param-record-layout>` `<sw-physic-type>` `<sw-unit>` `<sw-variable>` `<table>` `<team-member>` `<topic-1>` `<topic-2>` `<variant-char>` `<variant-def>` `<xdoc>` `<xfile>`

**Table 125: Attributes for SHORT-NAME**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Short designation, e.g. "TMOT". The `<short-name>` has an identifying character, in particular for outside references (e.g. *ASAP*, *ASCET*).

Example

## 17.8 SI-UNIT

`[SI-UNIT~]`-#PCDATA

**Figure 127: DTD-diagram for SI-UNIT**

Child elements none

parent elements `<sw-unit>`

**Table 126: Attributes for SI-UNIT**

Name	Type	Class	Value	Remark
[AMOUNT-OF-SUBSTANCE-EXPO]	cdata	implied		
[ELECTRIC-CURRENT-EXPO]	cdata	implied		
[LENGTH-EXPO]	cdata	implied		



**Table 126 (Cont.): Attributes for SI-UNIT**

Name	Type	Class	Value	Remark
[LUMINOUS-INTENSITY-EXPO]	cdata	implied		
[MASS-EXPO]	cdata	implied		
[S]	cdata	implied		
[THERMODYNAMIC-TEMPERATURE-EXPO]	cdata	implied		
[TIME-EXPO]	cdata	implied		

Description There are SI units of measure for the units of measure. STEP (ISO/DIS 10303-41. S96ff) is supported with regard to SI units by seven basic units (length, mass, time, electric\_current, thermodynamic\_temperature, amount\_of\_substance, luminous\_intensity). These basic units are realized by means of attributes.

Example

## 17.9 SPANSPEC

`<SPANSPEC>`-empty

**Figure 128: DTD-diagram for SPANSPEC**

Child elements none

parent elements `<tgroup>`

**Table 127: Attributes for SPANSPEC**

Name	Type	Class	Value	Remark
[ALIGN]	nmtkgrp	default	CENTER LEFT RIGHT CENTER JUSTIFY CHAR	
[CHAR]	cdata	implied		
[CHAROFF]	nutoken	implied		
[COLSEP]	number	implied		
[NAMEEND]	nmtoken	required		
[NAMEST]	nmtoken	required		
[ROWSEP]	number	implied		
[S]	cdata	implied		
[SPANNAME]	nmtoken	required		

## 17.10 STATE

`STATE~`—#PCDATA

Figure 129: DTD-diagram for STATE

Child elements none

parent elements `<company-revision-info>`

Table 128: Attributes for STATE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description

Example

## 17.11 STATE-1

`STATE-1~`—#PCDATA

Figure 130: DTD-diagram for STATE-1

Child elements none

parent elements `<std>` `<xdoc>`

Table 129: Attributes for STATE-1

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 17.12 STD

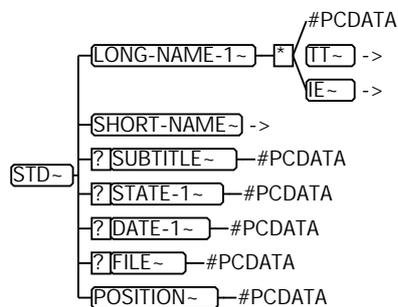


Figure 131: DTD-diagram for STD

Child elements `<long-name-1>` `<short-name>` `<subtitle>` `<state-1>` `<date-1>` `<file>` `<position>`

parent elements `<p>` `<td>`

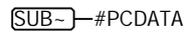
**Table 130: Attributes for STD**

Name	Type	Class	Value	Remark
[F-CHILD-TYPE]	cdata	fixed	date-1:date	
[F-ID-CLASS]	name	fixed	STD	
[ID]	id	required		
[S]	cdata	implied		

Description

Example

## 17.13 SUB



**Figure 132: DTD-diagram for SUB**

Child elements none

parent elements `<abs>` `<change>` `<desc>` `<ie>` `<indent-sample>` `<item-label>` `<max>` `<min>` `<p>` `<reason>` `<sw-unit-display>` `<tbr>` `<text>` `<tol>` `<typ>` `<unit>`

**Table 131: Attributes for SUB**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 17.14 SUBTITLE



**Figure 133: DTD-diagram for SUBTITLE**

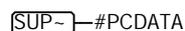
Child elements none

parent elements `<std>`

**Table 132: Attributes for SUBTITLE**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 17.15 SUP



**Figure 134: DTD-diagram for SUP**

Child elements none

parent elements `<abs>` `<change>` `<desc>` `<ie>` `<indent-sample>` `<item-label>` `<max>` `<min>` `<p>` `<reason>` `<sw-unit-display>` `<tbr>` `<text>` `<tol>` `<typ>` `<unit>`

Table 133: Attributes for SUP

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 17.16 SW-ADDRESSING-METHOD

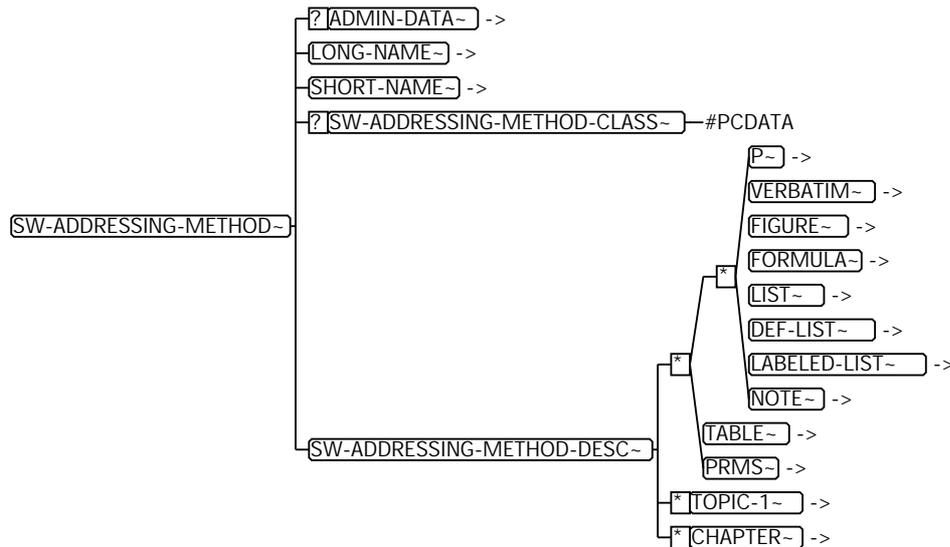


Figure 135: DTD-diagram for SW-ADDRESSING-METHOD

Child elements `<admin-data>` `<long-name>` `<short-name>` `<sw-addressing-method-class>` `<sw-addressing-method-desc>`

parent elements `<sw-addressing-methods>`

Table 134: Attributes for SW-ADDRESSING-METHOD

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	SW-ADDRESSING-METHOD	
[ID]	id	required		
[S]	cdata	implied		

**Description** Describes the addressing scheme. A description of how a parameter or a RAM variable from the control unit is given. Examples for the differing types of addressing are the direct addressing and the indirect addressing by a vector. The addressing scheme can also be used to define the RAM range in which the a RAM variable lies (e.g. in the internal or external RAM). Addressing schemes can be used by parameters and RAM variables.

This element is a reference to an access method for a name that must be simulated in the following systems (e.g. *adjustment systems*). This name links the MSR entity and the following systems.

Addressing schemes are not standardized or pre-defined in the DTD. Neither are they fully modeled. They are only described in terms of text in `<sw-addressing-method-desc>`.

Example

## 17.17 SW-ADDRESSING-METHOD-CLASS

`SW-ADDRESSING-METHOD-CLASS~`—#PCDATA

**Figure 136: DTD-diagram for SW-ADDRESSING-METHOD-CLASS**

Child elements none

parent elements `<sw-addressing-method>`

**Table 135: Attributes for SW-ADDRESSING-METHOD-CLASS**

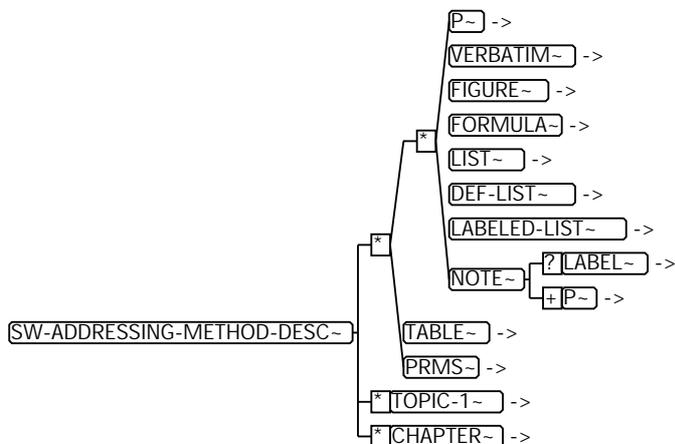
Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Gives the class for the addressing scheme. Parameters or variables can only use addressing schemes of a certain class. This however cannot be assured by SGML means; the application must do this.

Addressing-scheme classes are not standardized or pre-defined in the DTD.

Example

## 17.18 SW-ADDRESSING-METHOD-DESC



**Figure 137: DTD-diagram for SW-ADDRESSING-METHOD-DESC**

Child elements `<p>` `<verbatim>` `<figure>` `<formula>` `<list>` `<def-list>` `<labeled-list>` `<note>` `<table>` `<prms>` `<topic-1>` `<chapter>`

parent elements `<sw-addressing-method>`

**Table 136: Attributes for SW-ADDRESSING-METHOD-DESC**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Text-form description of the addressing schemes

Example

## 17.19 SW-ADDRESSING-METHOD-REF

`[SW-ADDRESSING-METHOD-REF~]` - #PCDATA

Figure 138: DTD-diagram for SW-ADDRESSING-METHOD-REF

Child elements none

parent elements `<sw-param>` `<sw-variable-implementation>`

Table 137: Attributes for SW-ADDRESSING-METHOD-REF

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND SW-ADDRESSING-METHOD	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		
[SW-ADDRESSING-METHOD]	idref	required		

Description Reference to the addressing scheme, e.g. by giving the short name.

Example

## 17.20 SW-ADDRESSING-METHODS

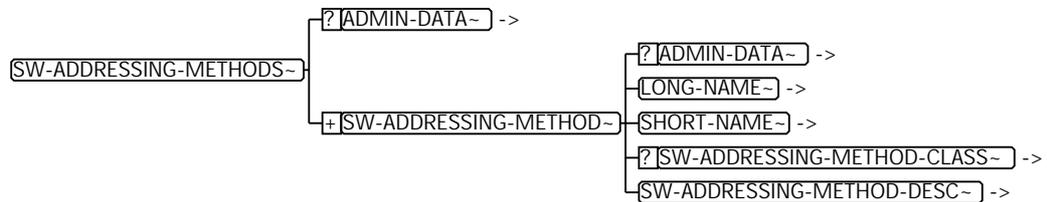


Figure 139: DTD-diagram for SW-ADDRESSING-METHODS

Child elements `<admin-data>` `<sw-addressing-method>`

parent elements `<sw-data-dictionary>`

Table 138: Attributes for SW-ADDRESSING-METHODS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description All addressing schemes are defined in `<sw-data-dictionary>`

Example

## 18 SW-APPLICATION-NOTES ... SW-COMPU-METHOD-TEXT-PAIR

### 18.1 SW-APPLICATION-NOTES

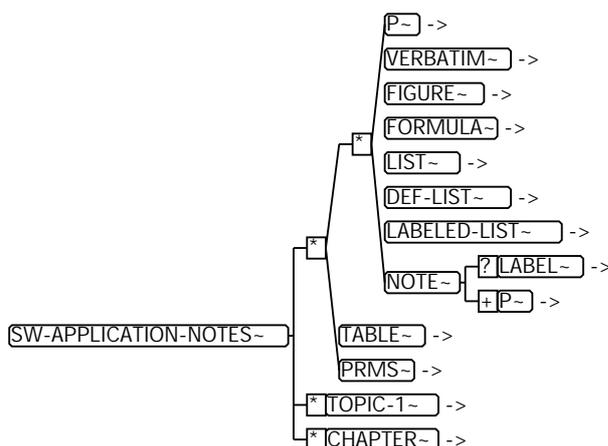


Figure 140: DTD-diagram for SW-APPLICATION-NOTES

Child elements `<p>` `<verbatim>` `<figure>` `<formula>` `<list>` `<def-list>` `<labeled-list>` `<note>`  
`<table>` `<prms>` `<topic-1>` `<chapter>`

parent elements `<sw-function-variant>`

Table 139: Attributes for SW-APPLICATION-NOTES

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Application notes

Example

### 18.2 SW-ARRAY-INDEX

`SW-ARRAY-INDEX~`—#PCDATA

Figure 141: DTD-diagram for SW-ARRAY-INDEX

Child elements none

parent elements `<sw-param-content>`

Table 140: Attributes for SW-ARRAY-INDEX

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Reference to the sub-structure for parameter structures.

Example

## 18.3 SW-ASAP-6-PRM-METHOD

~~SW-ASAP-6-PRM-METHOD~~ - #PCDATA

Figure 142: DTD-diagram for SW-ASAP-6-PRM-METHOD

Child elements none

parent elements <sw-compu-method>

Table 141: Attributes for SW-ASAP-6-PRM-METHOD

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description 6-parameter formula corresponding to ASAP (polynomial presentation. Presentation of the elements of the formula separated by white spaces. formula:  $int = (a \times x^2 + b \times x^1 + c) / (d \times x^2 + e \times x^1 + f)$

Example

## 18.4 SW-AXIS-GROUPED

~~SW-AXIS-GROUPED~~ - ~~SW-PARAM-REF~~ - #PCDATA

Figure 143: DTD-diagram for SW-AXIS-GROUPED

Child elements <sw-param-ref>

parent elements <sw-param-axis-x> <sw-param-axis-y>

Table 142: Attributes for SW-AXIS-GROUPED

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Several characteristics have the same datapoints for running-time reasons that are referenced by the name.

Example

## 18.5 SW-AXIS-INDIVIDUAL

~~SW-AXIS-INDIVIDUAL~~ - ~~SW-VARIABLE-REF~~ - #PCDATA  
~~SW-AXIS-INDIVIDUAL~~ - ~~MAX-COUNT~~ - #PCDATA  
~~SW-AXIS-INDIVIDUAL~~ - ~~SW-LIMITS~~ ->

Figure 144: DTD-diagram for SW-AXIS-INDIVIDUAL

Child elements <sw-variable-ref> <max-count> <sw-limits>

parent elements <sw-param-axis-x> <sw-param-axis-y> <sw-param-group-axis>

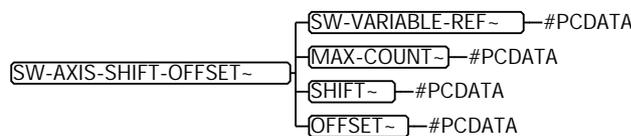
**Table 143: Attributes for SW-AXIS-INDIVIDUAL**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Description of an axis, the datapoints for which can be individually defined. Each characteristic or each map has its own private datapoint values (unlike a group characteristic).

Example

## 18.6 SW-AXIS-SHIFT-OFFSET



**Figure 145: DTD-diagram for SW-AXIS-SHIFT-OFFSET**

Child elements `<sw-variable-ref>` `<max-count>` `<shift>` `<offset>`

parent elements `<sw-param-axis-x>` `<sw-param-axis-y>`

**Table 144: Attributes for SW-AXIS-SHIFT-OFFSET**

Name	Type	Class	Value	Remark
[APPLICABLE]	number	default	0	
[S]	cdata	implied		

Description Special form of datapoints used for reasons of optimizing the running time. The datapoints are computed according to the following algorithm:  $\text{Datapoint}[i] = \dots + \text{Offset}$ .

Example

## 18.7 SW-BASE-TYPE



**Figure 146: DTD-diagram for SW-BASE-TYPE**

Child elements none

parent elements `<sw-param>` `<sw-variable-implementation>`

**Table 145: Attributes for SW-BASE-TYPE**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Base type of SW variable (e.g. char, integer).

Example

## 18.8 SW-BIT-REPRESENTATION

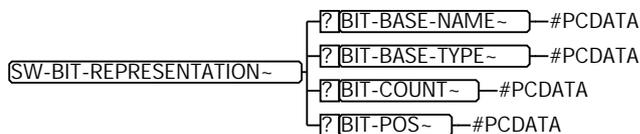


Figure 147: DTD-diagram for SW-BIT-REPRESENTATION

Child elements <bit-base-name> <bit-base-type> <bit-count> <bit-pos>

parent elements <sw-variable-implementation>

Table 146: Attributes for SW-BIT-REPRESENTATION

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description This optional element contains information or bit presentation in the form of elements (number of bits, bit position, bit base name, bit base type) for variables with the data type "bit".

Example

## 18.9 SW-CARB-DOC

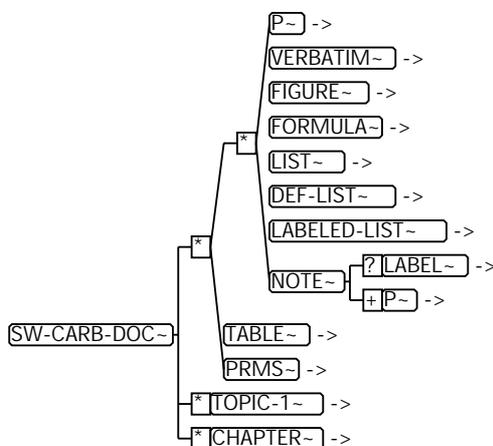


Figure 148: DTD-diagram for SW-CARB-DOC

Child elements <p> <verbatim> <figure> <formula> <list> <def-list> <labeled-list> <note> <table> <prms> <topic-1> <chapter>

parent elements <sw-function-variant>

Table 147: Attributes for SW-CARB-DOC

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description CARB documentation.

Example

## 18.10 SW-CODE-SYNTAX

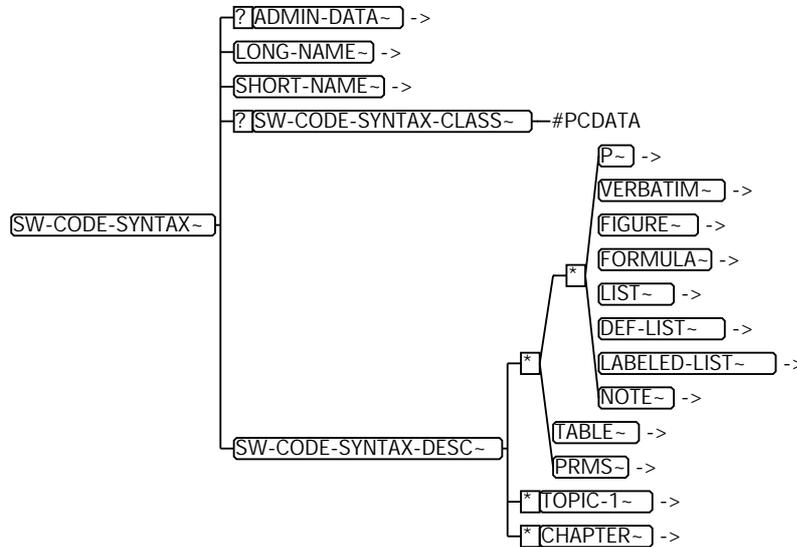


Figure 149: DTD-diagram for SW-CODE-SYNTAX

Child elements **<admin-data>** **<long-name>** **<short-name>** **<sw-code-syntax-class>** **<sw-code-syntax-desc>**

parent elements **<sw-code-syntaxes>**

Table 148: Attributes for SW-CODE-SYNTAX

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	SW-CODE-SYNTAX	
[ID]	id	required		
[S]	cdata	implied		

Description Name reference to a code syntax. The presentation of RAM variables and parameters in the data source file is defined by the code syntax. The code syntaxes are described in the **<sw-data-dictionary>**. The code syntaxes described here can be used by variables and parameters by referencing.

The content of a code syntax is not fully modeled, i.e. the code syntax is only described in text-form in **<sw-code-syntax-desc>**. There are no standardized values or values pre-defined in the DTD for the code syntax.

Example

## 18.11 SW-CODE-SYNTAX-CLASS

**SW-CODE-SYNTAX-CLASS~**—#PCDATA

Figure 150: DTD-diagram for SW-CODE-SYNTAX-CLASS

Child elements none

parent elements **<sw-code-syntax>**

**Table 149: Attributes for SW-CODE-SYNTAX-CLASS**

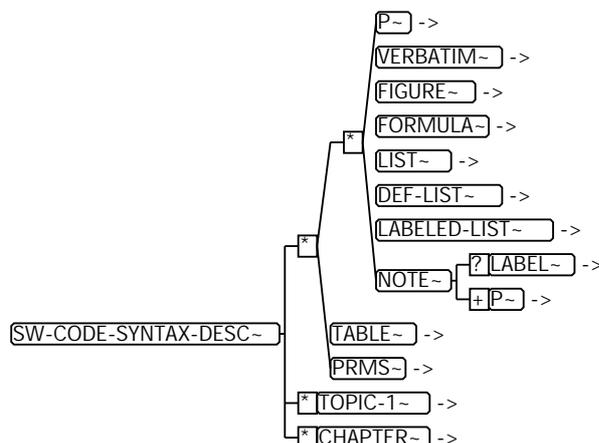
Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Gives the class for the code syntax. Parameters or variables can only use the code syntax of a certain class. This however cannot be assured by SGML means; the application must do this. A code syntax is for example, suitable for a parameter or for a variable, or only for a characteristic. The combination of a certain code syntax with a certain saving scheme can also be expressed by the class.

There are no standardized values or values pre-defined in the DTD for the code syntax classes.

Example

## 18.12 SW-CODE-SYNTAX-DESC



**Figure 151: DTD-diagram for SW-CODE-SYNTAX-DESC**

Child elements `<p>` `<verbatim>` `<figure>` `<formula>` `<list>` `<def-list>` `<labeled-list>` `<note>` `<table>` `<prms>` `<topic-1>` `<chapter>`

parent elements `<sw-code-syntax>`

**Table 150: Attributes for SW-CODE-SYNTAX-DESC**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Description of the code syntax

Example

## 18.13 SW-CODE-SYNTAX-REF

`SW-CODE-SYNTAX-REF` —#PCDATA

**Figure 152: DTD-diagram for SW-CODE-SYNTAX-REF**

Child elements none

parent elements <sw-param> <sw-variable-implementation>

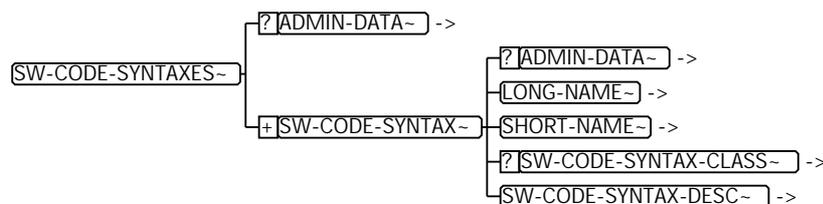
**Table 151: Attributes for SW-CODE-SYNTAX-REF**

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND SW-CODE-SYNTAX	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		
[SW-CODE-SYNTAX]	idref	required		

Description A reference to a code syntax can be made optionally within a RAM variable or parameter.

Example

## 18.14 SW-CODE-SYNTAXES



**Figure 153: DTD-diagram for SW-CODE-SYNTAXES**

Child elements <admin-data> <sw-code-syntax>

parent elements <sw-data-dictionary>

**Table 152: Attributes for SW-CODE-SYNTAXES**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description List of code syntaxes.

Example

## 18.15 SW-COMPU-GENERIC-MATH

[SW-COMPU-GENERIC-MATH~] #PCDATA

**Figure 154: DTD-diagram for SW-COMPU-GENERIC-MATH**

Child elements none

parent elements <sw-compu-method>

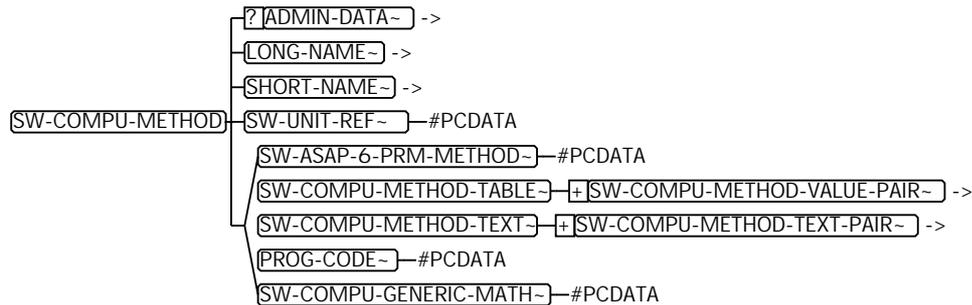
**Table 153: Attributes for SW-COMPU-GENERIC-MATH**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description The conversion formula is described by a general mathematical expression.

Example

## 18.16 SW-COMPU-METHOD



**Figure 155: DTD-diagram for SW-COMPU-METHOD**

Child elements `<admin-data>` `<long-name>` `<short-name>` `<sw-unit-ref>` `<sw-asap-6-prm-method>` `<sw-compu-method-table>` `<sw-compu-method-text>` `<prog-code>` `<sw-compu-generic-math>`

parent elements `<sw-compu-methods>`

**Table 154: Attributes for SW-COMPU-METHOD**

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	SW-COMPU-METHOD	
[ID]	id	required		
[S]	cdata	implied		

Description Conversion formula. Computing procedure according to which a control-unit-internal variable can be converted into its physical value. The conversion formula must be reversible.

Example

## 18.17 SW-COMPU-METHOD-REF

`SW-COMPU-METHOD-REF` cdata

**Figure 156: DTD-diagram for SW-COMPU-METHOD-REF**

Child elements none

parent elements `<sw-param-axis-values>` `<sw-variable-implementation>`

**Table 155: Attributes for SW-COMPU-METHOD-REF**

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND SW-COMPU-METHOD	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		
[SW-COMPU-METHOD]	idref	required		

Description Reference to a conversion formula

Example

## 18.18 SW-COMPU-METHOD-TABLE



**Figure 157: DTD-diagram for SW-COMPU-METHOD-TABLE**

Child elements `<sw-compu-method-value-pair>`

parent elements `<sw-compu-method>`

**Table 156: Attributes for SW-COMPU-METHOD-TABLE**

Name	Type	Class	Value	Remark
[INTERPOLATION-STYLE]	nmtkgrp	default	INTERPOLATION INTERPOLATION NO-INTERPOLATION DISCRETE	
[S]	cdata	implied		

Description With this form for the conversion formula, the relationship between internal and physical presentation is given using a table of for the internal-physical pairs of values. An attribute *interpolation-style* is given that can take the values *interpolation* (default) and *discrete*.

Example

## 18.19 SW-COMPU-METHOD-TEXT



**Figure 158: DTD-diagram for SW-COMPU-METHOD-TEXT**

Child elements `<sw-compu-method-text-pair>`

parent elements `<sw-compu-method>`

**Table 157: Attributes for SW-COMPU-METHOD-TEXT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description The conversion formula is described in text form. Unlike the other types of conversions, the physical side does not constitute a number but rather a text. Applications are conceivable for switches ("on"/"off") or for country identification ("D", "F", "US").

Example

## 18.20 SW-COMPU-METHOD-TEXT-PAIR



**Figure 159: DTD-diagram for SW-COMPU-METHOD-TEXT-PAIR**

Child elements `<cmt-int>` `<cmt-text>`

parent elements `<sw-compu-method-text>`

**Table 158: Attributes for SW-COMPU-METHOD-TEXT-PAIR**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Includes an internal and a text-form value for text-type conversion formulae.

Example

# 19 SW-COMPU-METHOD-VALUE-PAIR ... SW-FUNCTION-SPEC

## 19.1 SW-COMPU-METHOD-VALUE-PAIR



Figure 160: DTD-diagram for SW-COMPU-METHOD-VALUE-PAIR

Child elements `<cmt-int>` `<cmt-phys>`

parent elements `<sw-compu-method-table>`

Table 159: Attributes for SW-COMPU-METHOD-VALUE-PAIR

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description Includes an internal and a physical value for tabular types of conversion formulae.

Example

## 19.2 SW-COMPU-METHODS

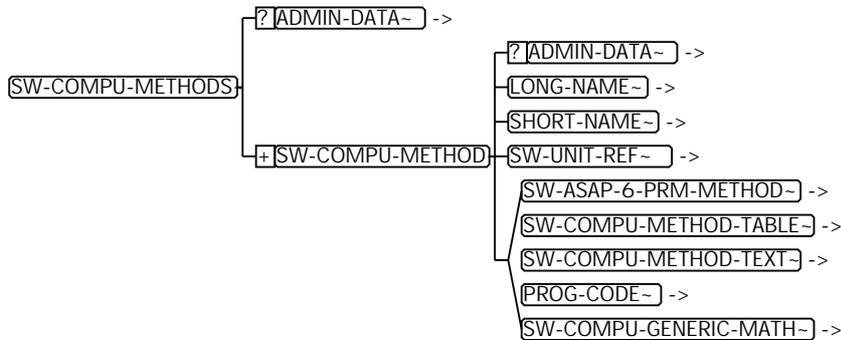


Figure 161: DTD-diagram for SW-COMPU-METHODS

Child elements `<admin-data>` `<sw-compu-method>`

parent elements `<sw-data-dictionary>`

Table 160: Attributes for SW-COMPU-METHODS

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description List of conversion formulae.

Example

## 19.3 SW-DATA-DICTIONARIES

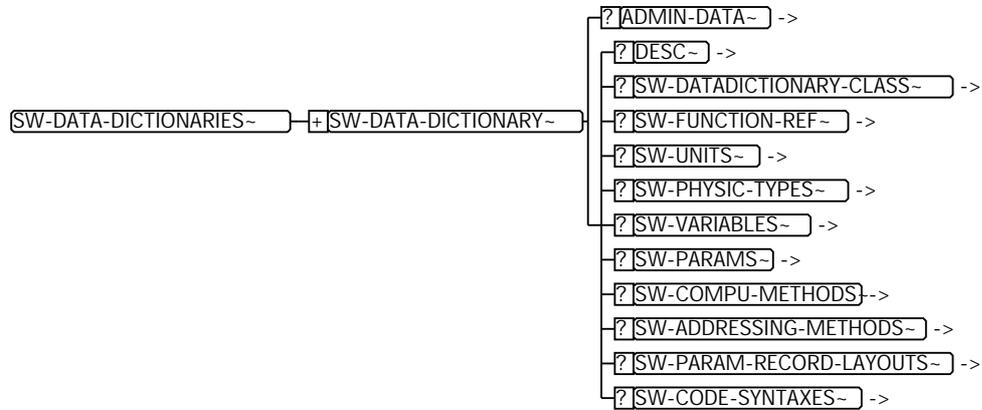


Figure 162: DTD-diagram for SW-DATA-DICTIONARIES

Child elements `<sw-data-dictionary>`

parent elements `<sw-data-dictionary-spec>`

Table 161: Attributes for SW-DATA-DICTIONARIES

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Size of the data dictionaries

Example

## 19.4 SW-DATA-DICTIONARY

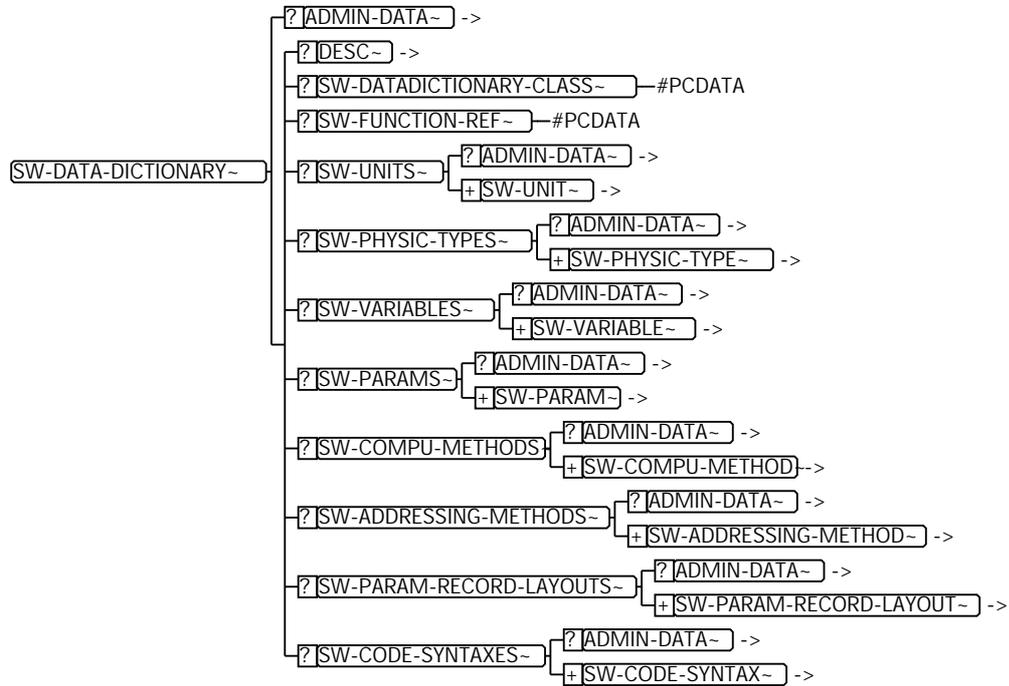


Figure 163: DTD-diagram for SW-DATA-DICTIONARY

Child elements `<admin-data>` `<desc>` `<sw-datadictionary-class>` `<sw-function-ref>` `<sw-units>` `<sw-physic-types>` `<sw-variables>` `<sw-params>` `<sw-compu-methods>` `<sw-addressing-methods>` `<sw-param-record-layouts>` `<sw-code-syntaxes>`

parent elements `<sw-data-dictionaries>`

Table 162: Attributes for SW-DATA-DICTIONARY

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Data dictionary. Data types, variables, parameters, units of measure, addressing schemes, storage schemes, code syntaxes and conversion formulae can be described here.

Example

## 19.5 SW-DATA-DICTIONARY-SPEC

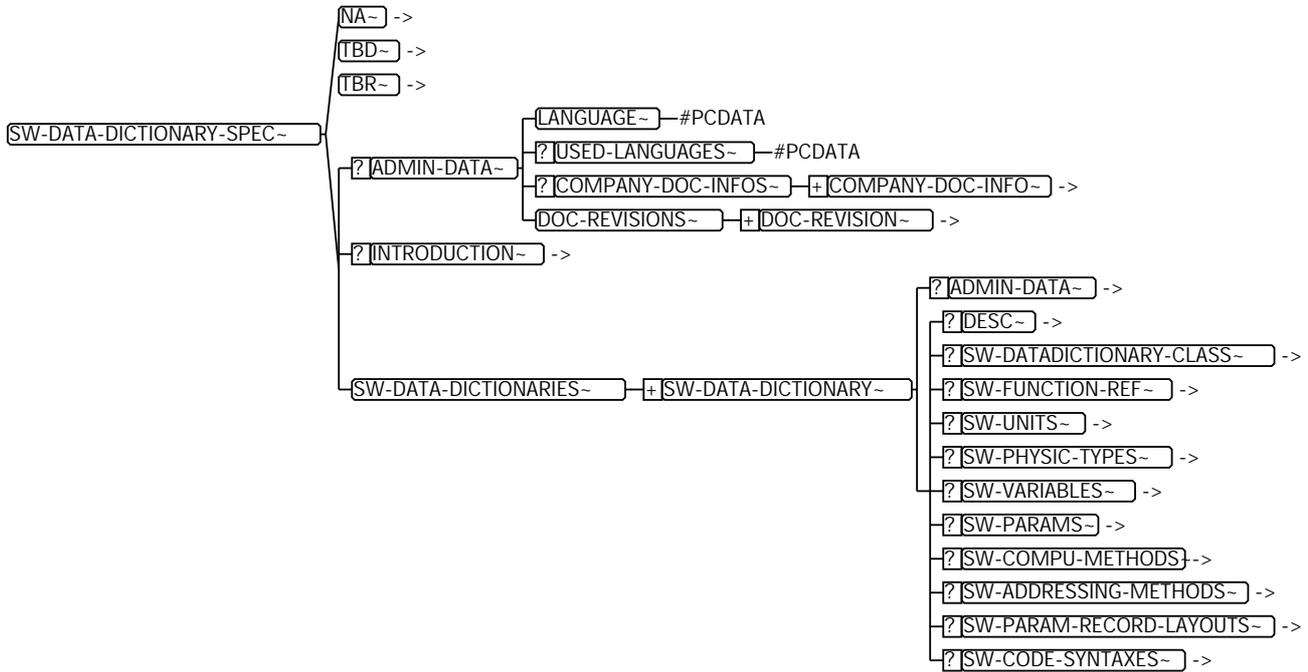


Figure 164: DTD-diagram for SW-DATA-DICTIONARY-SPEC

Child elements `<na>` `<tbd>` `<tbr>` `<admin-data>` `<introduction>` `<sw-data-dictionaries>`  
parent elements `<msrsw>` `<sw-function-variant>`

Table 163: Attributes for SW-DATA-DICTIONARY-SPEC

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description Data-dictionary specification.

Example

## 19.6 SW-DATA-TYPE-SCALING

`SW-DATA-TYPE-SCALING~` #PCDATA

Figure 165: DTD-diagram for SW-DATA-TYPE-SCALING

Child elements none

parent elements `<sw-physic-type>` `<sw-physic-type-1>`

Table 164: Attributes for SW-DATA-TYPE-SCALING

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description Quantization of a variable. Example: "16 bit".

Example

## 19.7 SW-DATADICTIONARY-CLASS

`[SW-DATADICTIONARY-CLASS~]`—#PCDATA

Figure 166: DTD-diagram for SW-DATADICTIONARY-CLASS

Child elements none

parent elements `<sw-data-dictionary>`

Table 165: Attributes for SW-DATADICTIONARY-CLASS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 19.8 SW-DECOMPOSITION

`[SW-DECOMPOSITION~]`+`[SW-FUNCTION-REF~]`—#PCDATA

Figure 167: DTD-diagram for SW-DECOMPOSITION

Child elements `<sw-function-ref>`

parent elements `<sw-function-variant>`

Table 166: Attributes for SW-DECOMPOSITION

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Formation of functional hierarchies by referencing included and used functions.

Example

## 19.9 SW-FULFILS

`[SW-FULFILS~]`+`[FUNCTION-REF~]`—#PCDATA

Figure 168: DTD-diagram for SW-FULFILS

Child elements `<function-ref>`

parent elements `<sw-function-variant>`

Table 167: Attributes for SW-FULFILS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 19.10 SW-FUNCTION

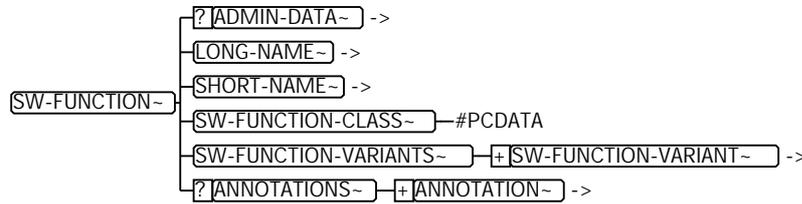


Figure 169: DTD-diagram for SW-FUNCTION

Child elements **<admin-data>** **<long-name>** **<short-name>** **<sw-function-class>** **<sw-function-variants>** **<annotations>**

parent elements **<sw-functions>**

Table 168: Attributes for SW-FUNCTION

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	SW-FUNCTION	
[F-NAMESPACE]	names	fixed	PRM	
[ID]	id	required		
[S]	cdata	implied		

Description Software function

Example

## 19.11 SW-FUNCTION-CLASS



Figure 170: DTD-diagram for SW-FUNCTION-CLASS

Child elements none

parent elements **<sw-function>**

Table 169: Attributes for SW-FUNCTION-CLASS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Function class. The functions can be broken down into different classes. There can be functions for example occurring only in the documentation/analysis phase that can be grouped to a class.

Example

## 19.12 SW-FUNCTION-DEF



Figure 171: DTD-diagram for SW-FUNCTION-DEF

Child elements **<figure>** **<ncoi-1>**

parent elements **<sw-function-variant>**

Table 170: Attributes for SW-FUNCTION-DEF

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description A short decription for each function is given by a suitable means of presentation. A graphical and a short text are used as a rule. Example: ASCET function model, status diagrams, SA/SD diagrams.

Example

## 19.13 SW-FUNCTION-DESC

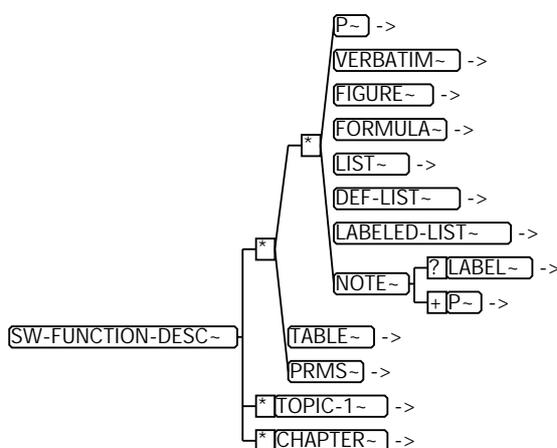


Figure 172: DTD-diagram for SW-FUNCTION-DESC

Child elements **<p>** **<verbatim>** **<figure>** **<formula>** **<list>** **<def-list>** **<labeled-list>** **<note>** **<table>** **<prms>** **<topic-1>** **<chapter>**

parent elements **<sw-function-variant>**

Table 171: Attributes for SW-FUNCTION-DESC

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Extensive description in text form for the function.

Example

## 19.14 SW-FUNCTION-EXPORT-VARIABLES

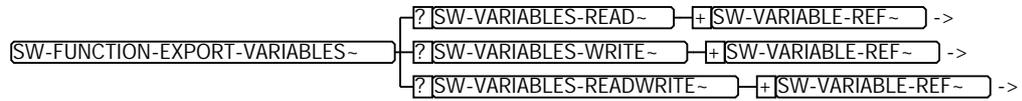


Figure 173: DTD-diagram for SW-FUNCTION-EXPORT-VARIABLES

Child elements **<sw-variables-read>** **<sw-variables-write>** **<sw-variables-readwrite>**

parent elements **<sw-function-variables>**

Table 172: Attributes for SW-FUNCTION-EXPORT-VARIABLES

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description List of exported variables. When a function defines a variable, i.e., provides a variable or the value, then it is to be listed here.

Example

## 19.15 SW-FUNCTION-IMPORT-VARIABLES

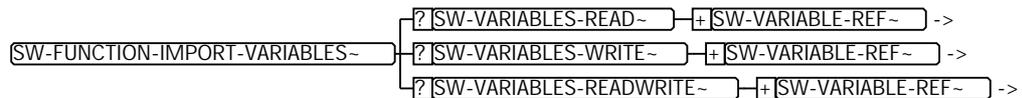


Figure 174: DTD-diagram for SW-FUNCTION-IMPORT-VARIABLES

Child elements **<sw-variables-read>** **<sw-variables-write>** **<sw-variables-readwrite>**

parent elements **<sw-function-variables>**

Table 173: Attributes for SW-FUNCTION-IMPORT-VARIABLES

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description List of imported variables. Variables are listed here that are made available by other functions.

Example

## 19.16 SW-FUNCTION-LOCAL-VARIABLES



Figure 175: DTD-diagram for SW-FUNCTION-LOCAL-VARIABLES

Child elements **<sw-variables-readwrite>**

parent elements **<sw-function-variables>**



**Table 174: Attributes for SW-FUNCTION-LOCAL-VARIABLES**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description List of local variables, i.e. no transfer variable.

Example

## 19.17 SW-FUNCTION-MODELONLY-VARIABLES

`[SW-FUNCTION-MODELONLY-VARIABLES~]` `+ [SW-VARIABLE-REF~]` `#PCDATA`

**Figure 176: DTD-diagram for SW-FUNCTION-MODELONLY-VARIABLES**

Child elements `<sw-variable-ref>`

parent elements `<sw-function-variables>`

**Table 175: Attributes for SW-FUNCTION-MODELONLY-VARIABLES**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description List of variables that only exist in the model and no longer occur during implementation.

Example

## 19.18 SW-FUNCTION-REF

`[SW-FUNCTION-REF~]` `#PCDATA`

**Figure 177: DTD-diagram for SW-FUNCTION-REF**

Child elements none

parent elements `<sw-data-dictionary>` `<sw-decomposition>` `<sw-function-refs>` `<sw-param-content>` `<sw-param-contents>`

**Table 176: Attributes for SW-FUNCTION-REF**

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND SW-FUNCTION	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		
[SW-FUNCTION]	idref	required		

Description In `<msrsw>`, reference to functions used that are described outside of the SW. In `<sw-decomposition>`, reference to functions that are described within `<msrsw>`.

Example

## 19.19 SW-FUNCTION-REFS

`[SW-FUNCTION-REFS~]` `+ [SW-FUNCTION-REF~]` `- #PCDATA`

Figure 178: DTD-diagram for SW-FUNCTION-REFS

Child elements `<sw-function-ref>`

parent elements `<msrsw>`

Table 177: Attributes for SW-FUNCTION-REFS

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description References to functions used that are described outside of the document, e.g. operating system functions, network services.

Example

## 19.20 SW-FUNCTION-SPEC

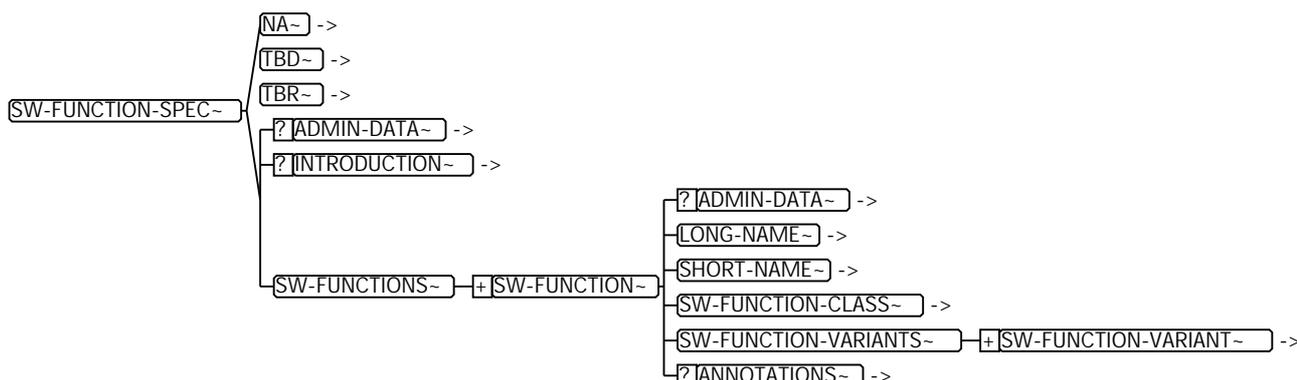


Figure 179: DTD-diagram for SW-FUNCTION-SPEC

Child elements `<na>` `<tbd>` `<tbr>` `<admin-data>` `<introduction>` `<sw-functions>`

parent elements `<msrsw>`

Table 178: Attributes for SW-FUNCTION-SPEC

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description Functions specification. A functions specification can include several functions. The functions can be in hierarchical form by which reference to a function is made by `<sw-decomposition>`.

Example

## 20 SW-FUNCTION-VARIABLES ... SW-PARAM-CONTENT-Y

### 20.1 SW-FUNCTION-VARIABLES

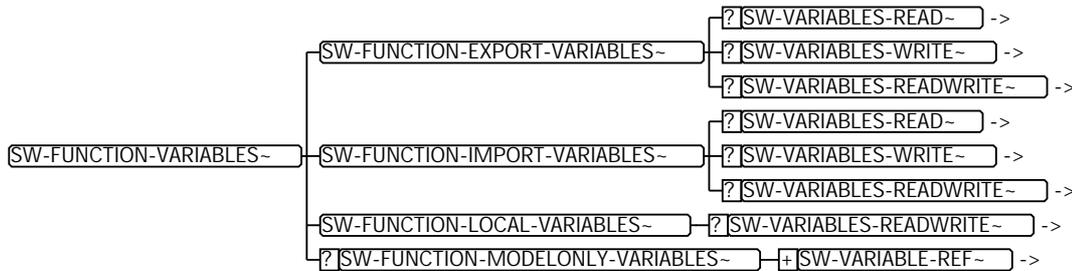


Figure 180: DTD-diagram for SW-FUNCTION-VARIABLES

Child elements **<sw-function-export-variables>** **<sw-function-import-variables>** **<sw-function-local-variables>** **<sw-function-modelonly-variables>**

parent elements **<sw-function-variant>**

Table 179: Attributes for SW-FUNCTION-VARIABLES

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description List of variables used by the function

Example

## 20.2 SW-FUNCTION-VARIANT

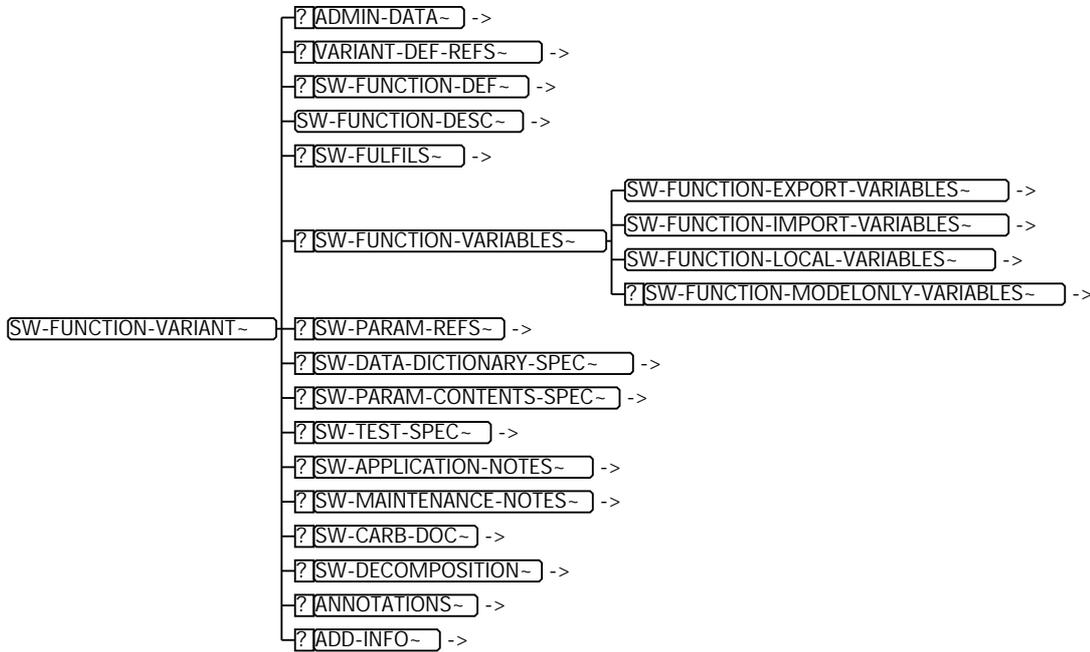


Figure 181: DTD-diagram for SW-FUNCTION-VARIANT

Child elements **<admin-data>** **<variant-def-refs>** **<sw-function-def>** **<sw-function-desc>**  
**<sw-fulfils>** **<sw-function-variables>** **<sw-param-refs>** **<sw-data-dictionary-spec>**  
**<sw-param-contents-spec>** **<sw-test-spec>** **<sw-application-notes>** **<sw-maintenance-notes>**  
**<sw-carb-doc>** **<sw-decomposition>** **<annotations>** **<add-info>**

parent elements **<sw-function-variants>**

Table 180: Attributes for SW-FUNCTION-VARIANT

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Function variant. If no function variants are used, then a minimum of one still has to be given. Each function variant must be completely described in the implementation V1.1.0, i.e. there is no re-use.

A features bar in **<project-data>** ... **<var-def>** (function switch) can be referenced by means of **<variant-def-ref>**.

Example

## 20.3 SW-FUNCTION-VARIANTS

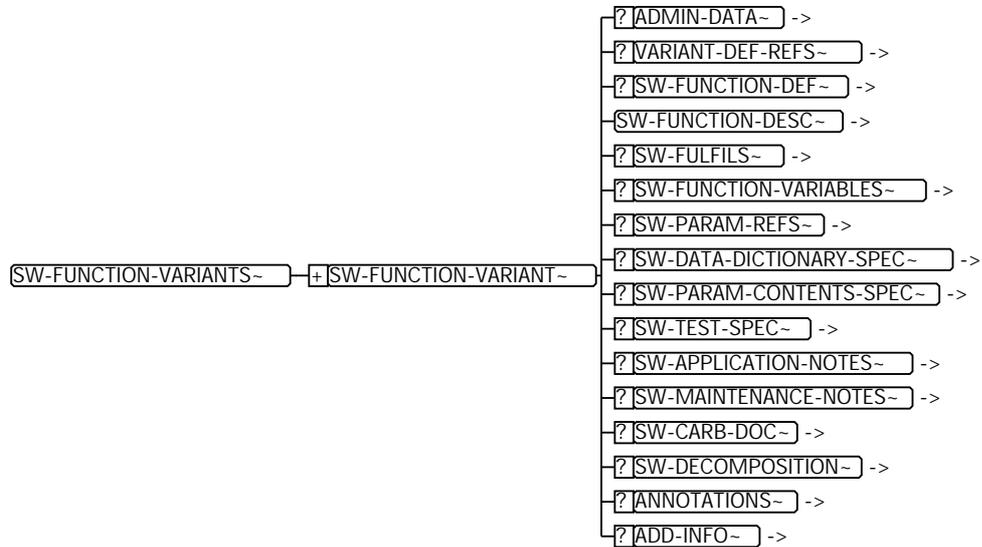


Figure 182: DTD-diagram for SW-FUNCTION-VARIANTS

Child elements **<sw-function-variant>**

parent elements **<sw-function>**

Table 181: Attributes for SW-FUNCTION-VARIANTS

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description Number of function variants.

Example

## 20.4 SW-FUNCTIONS

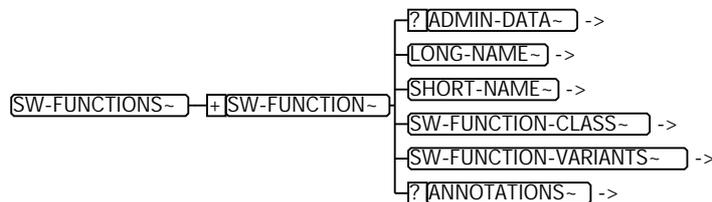


Figure 183: DTD-diagram for SW-FUNCTIONS

Child elements **<sw-function>**

parent elements **<sw-function-spec>**

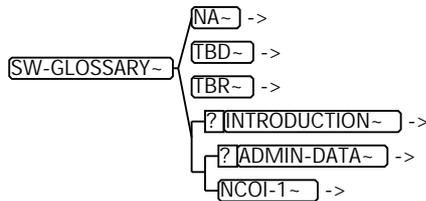
**Table 182: Attributes for SW-FUNCTIONS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Number of functions in the functional specification/in the document. Envelope element.

Example

## 20.5 SW-GLOSSARY



**Figure 184: DTD-diagram for SW-GLOSSARY**

Child elements `<na>` `<tbd>` `<tbr>` `<introduction>` `<admin-data>` `<ncoi-1>`

parent elements `<msrsw>`

**Table 183: Attributes for SW-GLOSSARY**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Glossary

Example

## 20.6 SW-INTERPOLATION-METHOD

`SW-INTERPOLATION-METHOD~`—#PCDATA

**Figure 185: DTD-diagram for SW-INTERPOLATION-METHOD**

Child elements none

parent elements `<sw-param>`

**Table 184: Attributes for SW-INTERPOLATION-METHOD**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Interpolation method for parameters. The methods are not standardized or defined in the DTD. The methods are described in text form.

Example

## 20.7 SW-LIMITS

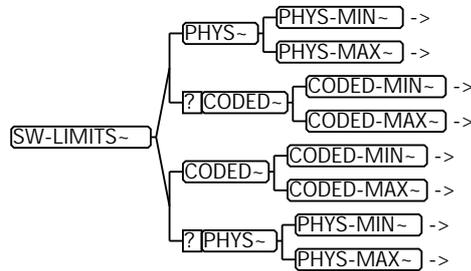


Figure 186: DTD-diagram for SW-LIMITS

Child elements **<phys>** **<coded>** **<coded>** **<phys>**

parent elements **<sw-axis-individual>** **<sw-param-axis-values>** **<sw-physic-type>** **<sw-physic-type-1>** **<sw-variable-implementation>**

Table 185: Attributes for SW-LIMITS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Minimum and maximum values for the physical and the control-unit-internal values are given here.

Example

## 20.8 SW-MAINTENANCE-NOTES

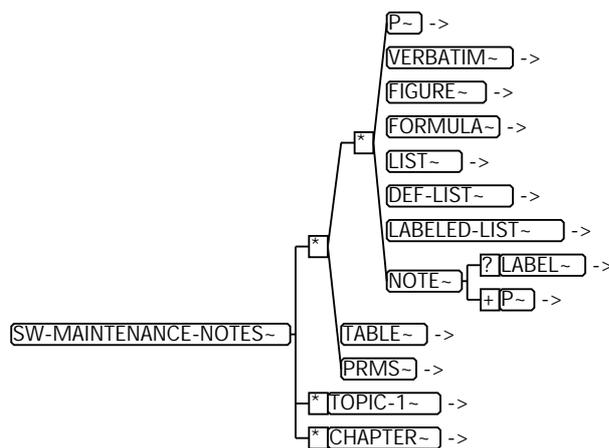


Figure 187: DTD-diagram for SW-MAINTENANCE-NOTES

Child elements **<p>** **<verbatim>** **<figure>** **<formula>** **<list>** **<def-list>** **<labeled-list>** **<note>** **<table>** **<prms>** **<topic-1>** **<chapter>**

parent elements **<sw-function-variant>**

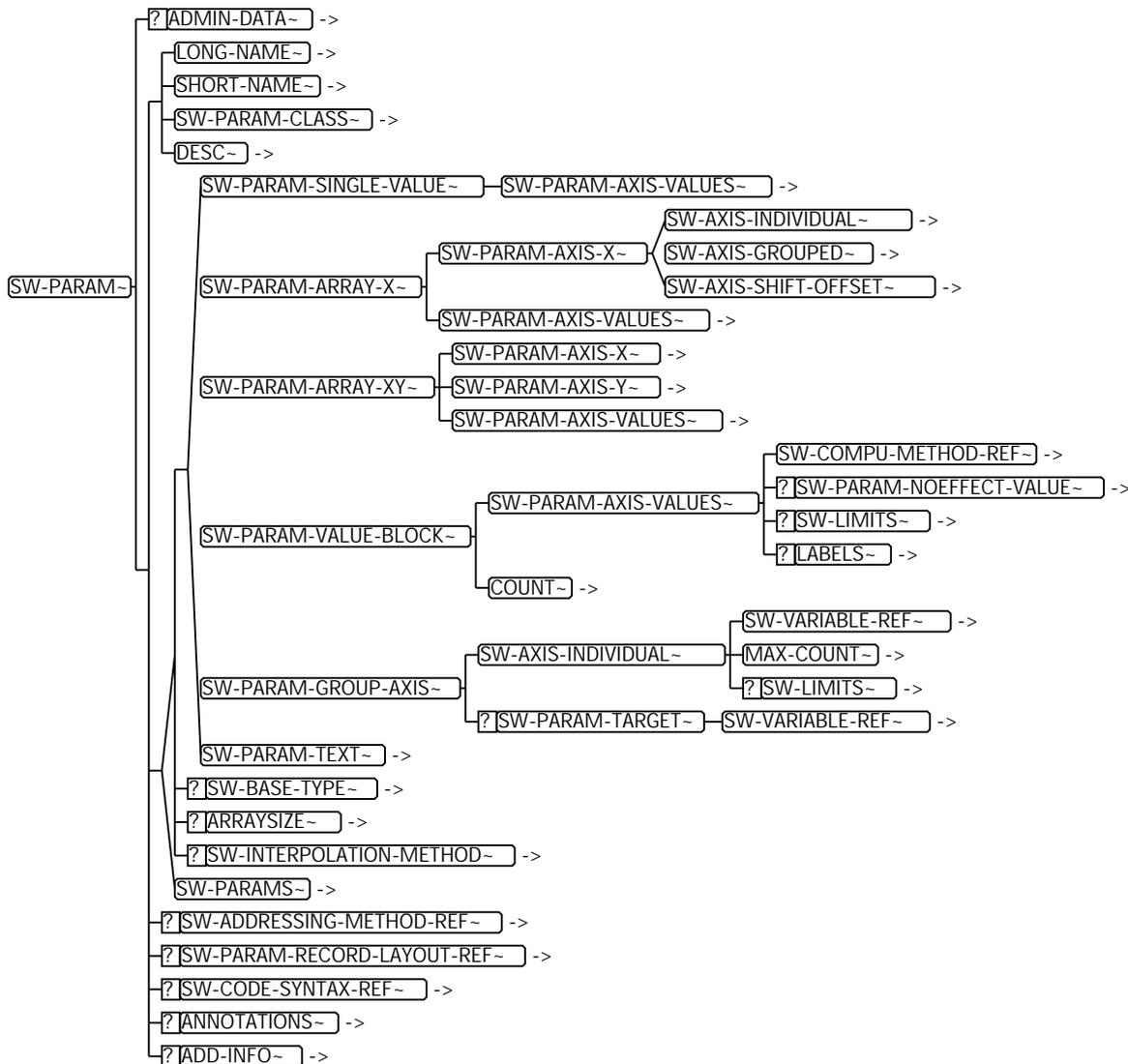
**Table 186: Attributes for SW-MAINTENANCE-NOTES**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Customer servicing notes

Example

## 20.9 SW-PARAM



**Figure 188: DTD-diagram for SW-PARAM**

Child elements `<admin-data>` `<long-name>` `<short-name>` `<sw-param-class>` `<desc>`  
`<sw-param-single-value>` `<sw-param-array-x>` `<sw-param-array-xy>`  
`<sw-param-value-block>` `<sw-param-group-axis>` `<sw-param-text>` `<sw-base-type>` `<arraysize>` `<sw-interpolation-method>` `<sw-params>` `<sw-addressing-method-ref>` `<sw-param-record-layout-ref>` `<sw-code-syntax-ref>` `<annotations>` `<add-info>`

parent elements **<sw-params>**

**Table 187: Attributes for SW-PARAM**

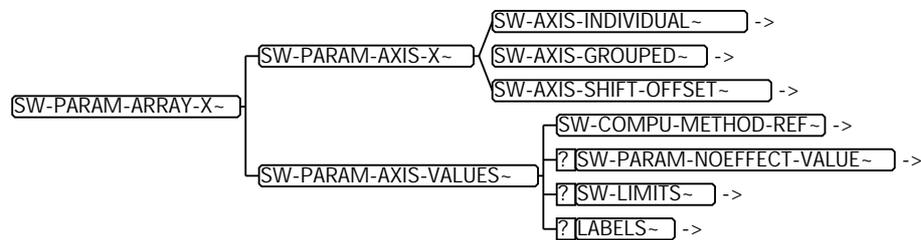
Name	Type	Class	Value	Remark
[CALIBRATION]	nmtkgrp	default	CALIBRATION NOT-IN-MEMORY CALIBRATION NO-CALIBRATION	
[F-ID-CLASS]	name	fixed	SW-PARAM	
[ID]	id	required		
[S]	cdata	implied		

Description Software parameters. Software parameters are parameters ( in ASAP characteristics): characteristics, maps and fixed values.

Parameters are as a rule, stored in the EPROM of the control unit. They are applicable, i.e. they can be read or modified using application tools within the scope of the plausibility and within the scope of set access protection attributes. Exceptions: e.g. cylinder number, not in the EPROM, not applicable. There is one attribute *calibration* for the possible attribute values *calibration* (default), *no-calibration* and *not-in-memory*.

Example

## 20.10 SW-PARAM-ARRAY-X



**Figure 189: DTD-diagram for SW-PARAM-ARRAY-X**

Child elements **<sw-param-axis-x>** **<sw-param-axis-values>**

parent elements **<sw-param>**

**Table 188: Attributes for SW-PARAM-ARRAY-X**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Characteristic

Example

## 20.11 SW-PARAM-ARRAY-XY

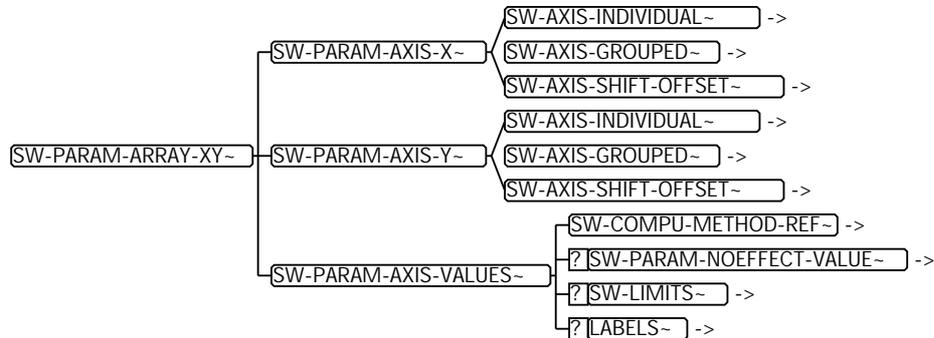


Figure 190: DTD-diagram for SW-PARAM-ARRAY-XY

Child elements `<sw-param-axis-x>` `<sw-param-axis-y>` `<sw-param-axis-values>`

parent elements `<sw-param>`

Table 189: Attributes for SW-PARAM-ARRAY-XY

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Map

Example

## 20.12 SW-PARAM-AXIS-VALUES

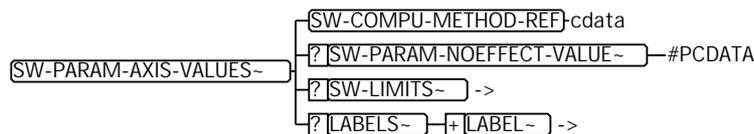


Figure 191: DTD-diagram for SW-PARAM-AXIS-VALUES

Child elements `<sw-compu-method-ref>` `<sw-param-noeffect-value>` `<sw-limits>` `<labels>`

parent elements `<sw-param-array-x>` `<sw-param-array-xy>` `<sw-param-single-value>` `<sw-param-value-block>`

Table 190: Attributes for SW-PARAM-AXIS-VALUES

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Value axis. Description of a characteristic value or the values of a characteristic or map

Example

## 20.13 SW-PARAM-AXIS-X

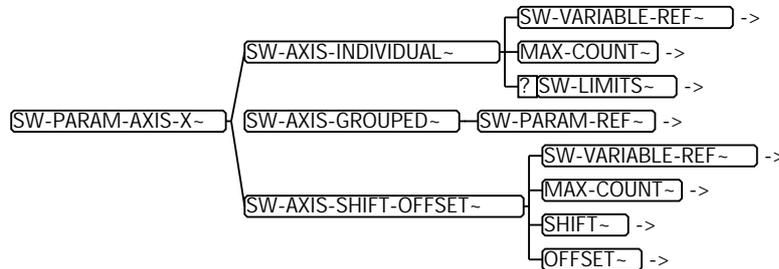


Figure 192: DTD-diagram for SW-PARAM-AXIS-X

Child elements **<sw-axis-individual>** **<sw-axis-grouped>** **<sw-axis-shift-offset>**

parent elements **<sw-param-array-x>** **<sw-param-array-xy>**

Table 191: Attributes for SW-PARAM-AXIS-X

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Describes the x axis of a parameter (**<sw-param>**). This description consists of the input variable (as reference to a variable), min, max, max-count (max. number of datapoints).

Example

## 20.14 SW-PARAM-AXIS-Y

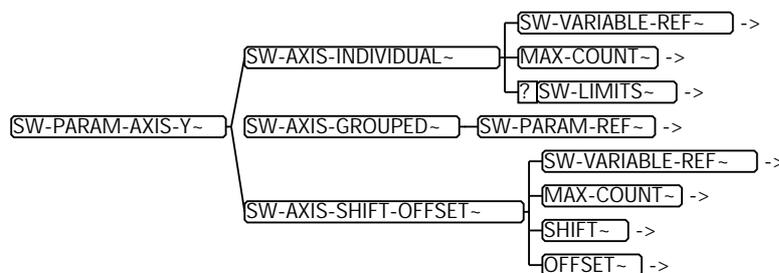


Figure 193: DTD-diagram for SW-PARAM-AXIS-Y

Child elements **<sw-axis-individual>** **<sw-axis-grouped>** **<sw-axis-shift-offset>**

parent elements **<sw-param-array-xy>**

Table 192: Attributes for SW-PARAM-AXIS-Y

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Describes the y axis of a parameter (**<sw-param>**). This description consists of the input variable (as reference to a variable), min, max, max-count (max. number of datapoints).

Example

## 20.15 SW-PARAM-CLASS

`[SW-PARAM-CLASS~]`—#PCDATA

Figure 194: DTD-diagram for SW-PARAM-CLASS

Child elements none

parent elements `<sw-param>` `<sw-param-content>`

Table 193: Attributes for SW-PARAM-CLASS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Gives the type of parameter (characteristic, map, characteristic value, fixed characteristic, group characteristic, fixed map).

Example

## 20.16 SW-PARAM-CONTENT

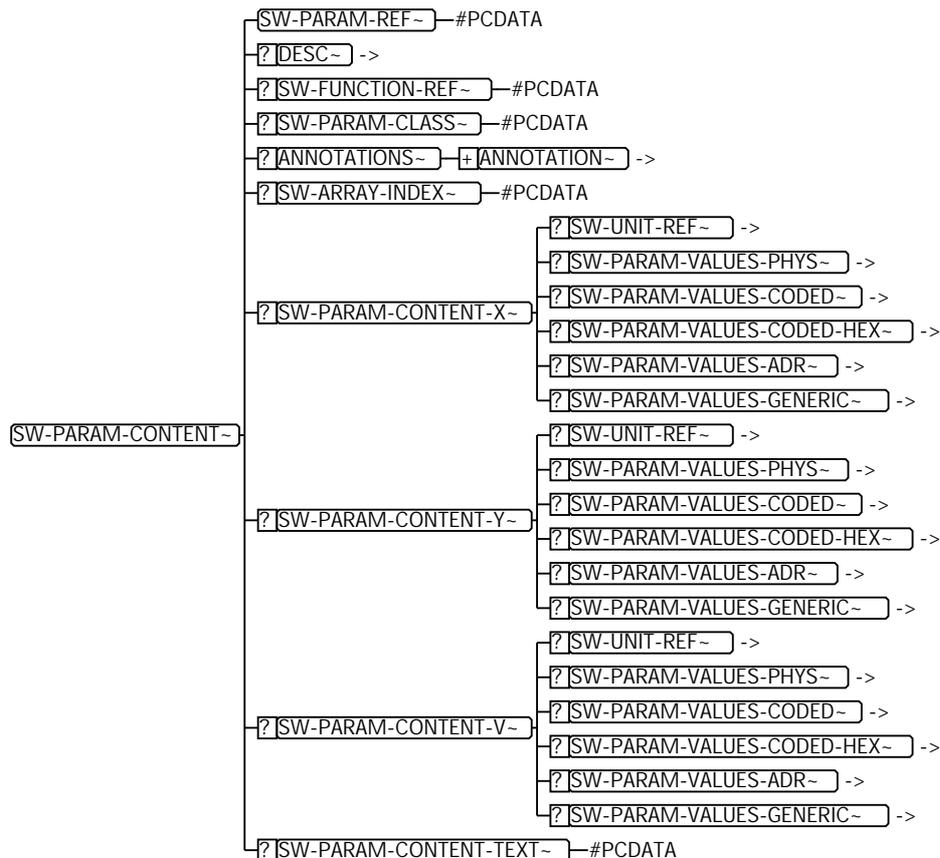


Figure 195: DTD-diagram for SW-PARAM-CONTENT

Child elements `<sw-param-ref>` `<desc>` `<sw-function-ref>` `<sw-param-class>` `<annotations>` `<sw-array-index>` `<sw-param-content-x>` `<sw-param-content-y>` `<sw-param-content-v>` `<sw-param-content-text>`

parent elements **<sw-param-contents>**

**Table 194: Attributes for SW-PARAM-CONTENT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description Parameter values

Example

## 20.17 SW-PARAM-CONTENT-TEXT

`[SW-PARAM-CONTENT-TEXT~]`—#PCDATA

**Figure 196: DTD-diagram for SW-PARAM-CONTENT-TEXT**

Child elements none

parent elements **<sw-param-content>**

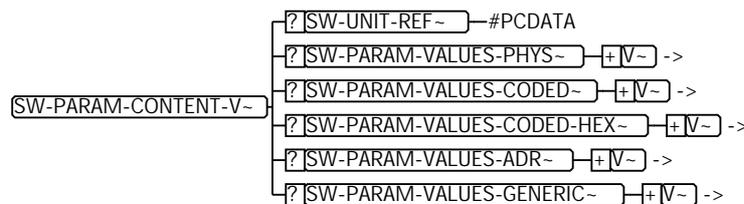
**Table 195: Attributes for SW-PARAM-CONTENT-TEXT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description The values of a test parameter are given here (sw-param-text).

Example

## 20.18 SW-PARAM-CONTENT-V



**Figure 197: DTD-diagram for SW-PARAM-CONTENT-V**

Child elements **<sw-unit-ref>** **<sw-param-values-phys>** **<sw-param-values-coded>** **<sw-param-values-coded-hex>** **<sw-param-values-adr>** **<sw-param-values-generic>**

parent elements **<sw-param-content>**

**Table 196: Attributes for SW-PARAM-CONTENT-V**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

Description An individual value is given here for a characteristic value. The values of the v axis are given here for a characteristic. The values of the v axis are given here for a map. The following convention thereby applies: Index of the x axis runs faster. Values are separated by blanks or line feeds.

Example

## 20.19 SW-PARAM-CONTENT-X

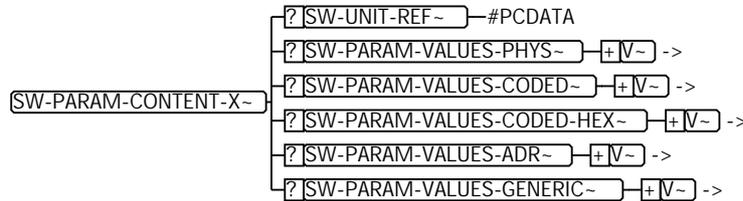


Figure 198: DTD-diagram for SW-PARAM-CONTENT-X

Child elements **<sw-unit-ref>** **<sw-param-values-phys>** **<sw-param-values-coded>** **<sw-param-values-coded-hex>** **<sw-param-values-adr>** **<sw-param-values-generic>**

parent elements **<sw-param-content>**

Table 197: Attributes for SW-PARAM-CONTENT-X

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description The physical, internal, ... values for the x axis are given here.

Example

## 20.20 SW-PARAM-CONTENT-Y

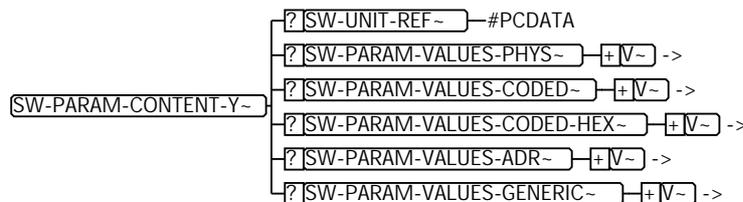


Figure 199: DTD-diagram for SW-PARAM-CONTENT-Y

Child elements **<sw-unit-ref>** **<sw-param-values-phys>** **<sw-param-values-coded>** **<sw-param-values-coded-hex>** **<sw-param-values-adr>** **<sw-param-values-generic>**

parent elements **<sw-param-content>**

Table 198: Attributes for SW-PARAM-CONTENT-Y

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

Description The physical, internal, ... values for the y axis are given here.

Example

## 21 SW-PARAM-CONTENTS ... SW-PARAM-VALUES-GENERIC

### 21.1 SW-PARAM-CONTENTS

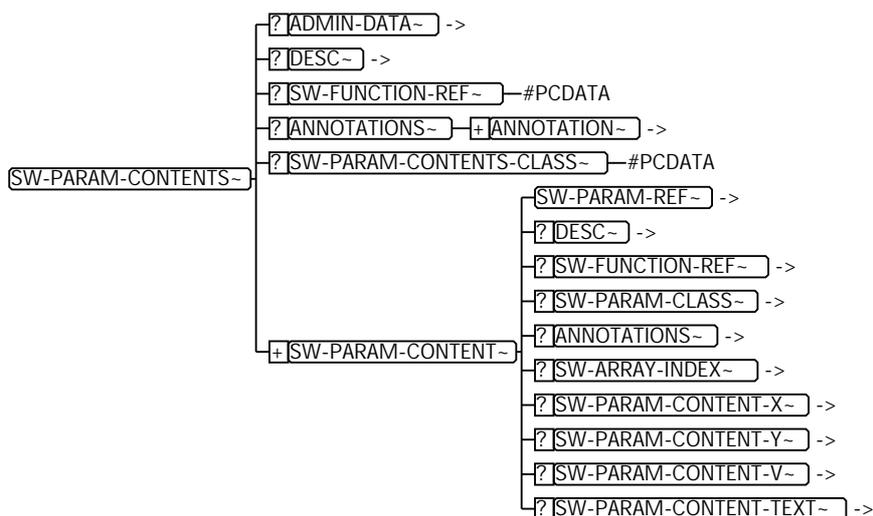


Figure 200: DTD-diagram for SW-PARAM-CONTENTS

Child elements `<admin-data>` `<desc>` `<sw-function-ref>` `<annotations>` `<sw-param-contents-class>` `<sw-param-content>`

parent elements `<sw-param-contents-spec>`

Table 199: Attributes for SW-PARAM-CONTENTS

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

### 21.2 SW-PARAM-CONTENTS-CLASS

`SW-PARAM-CONTENTS-CLASS~` -#PCDATA

Figure 201: DTD-diagram for SW-PARAM-CONTENTS-CLASS

Child elements none

parent elements `<sw-param-contents>`

Table 200: Attributes for SW-PARAM-CONTENTS-CLASS

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

## 21.3 SW-PARAM-CONTENTS-SPEC

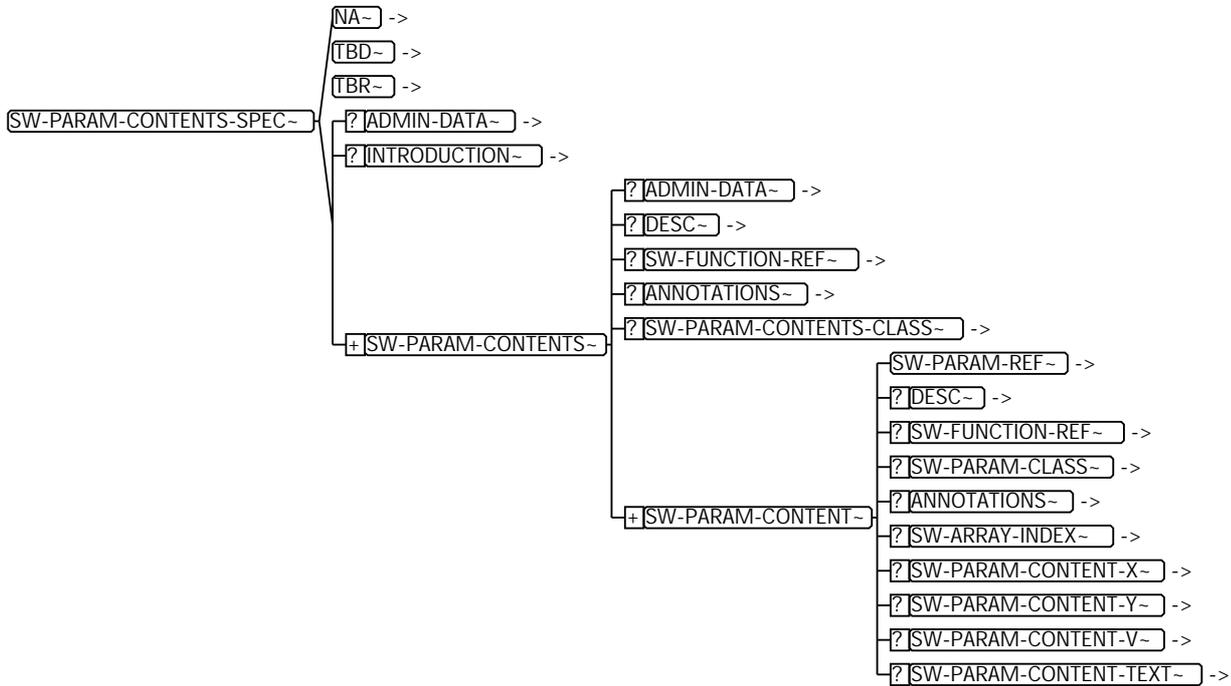


Figure 202: DTD-diagram for SW-PARAM-CONTENTS-SPEC

Child elements <na> <tbd> <tbr> <admin-data> <introduction> <sw-param-contents>

parent elements <msrsw> <sw-function-variant>

Table 201: Attributes for SW-PARAM-CONTENTS-SPEC

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 21.4 SW-PARAM-GROUP-AXIS

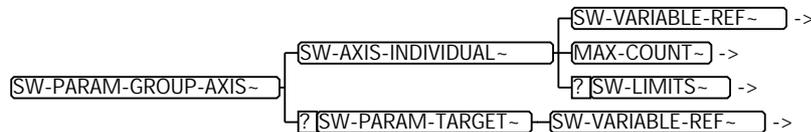


Figure 203: DTD-diagram for SW-PARAM-GROUP-AXIS

Child elements <sw-axis-individual> <sw-param-target>

parent elements <sw-param>

Table 202: Attributes for SW-PARAM-GROUP-AXIS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 21.5 SW-PARAM-NOEFFECT-VALUE

`[SW-PARAM-NOEFFECT-VALUE~]` -#PCDATA

Figure 204: DTD-diagram for SW-PARAM-NOEFFECT-VALUE

Child elements none

parent elements `<sw-param-axis-values>`

Table 203: Attributes for SW-PARAM-NOEFFECT-VALUE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 21.6 SW-PARAM-RECORD-LAYOUT

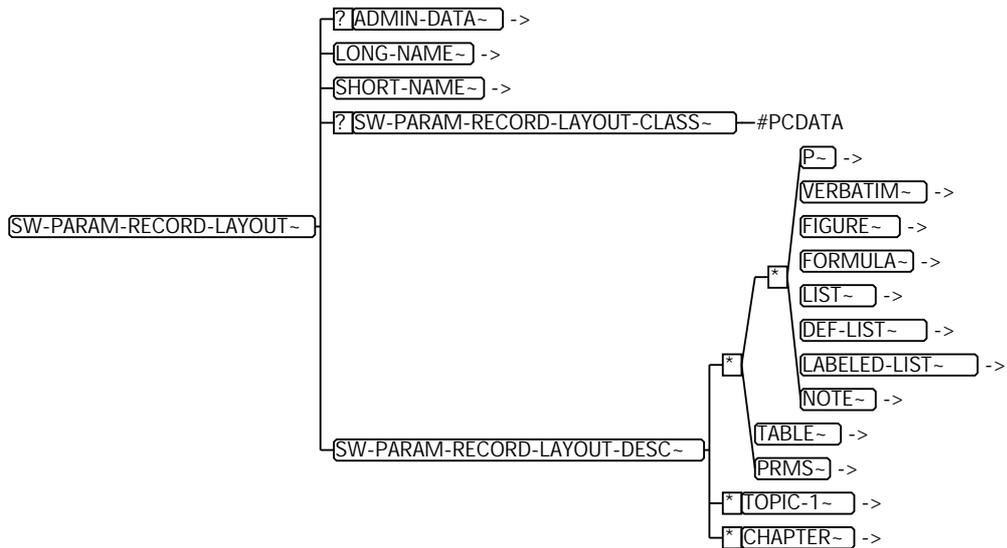


Figure 205: DTD-diagram for SW-PARAM-RECORD-LAYOUT

Child elements `<admin-data>` `<long-name>` `<short-name>` `<sw-param-record-layout-class>` `<sw-param-record-layout-desc>`

parent elements `<sw-param-record-layouts>`

Table 204: Attributes for SW-PARAM-RECORD-LAYOUT

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	SW-PARAM-RECORD-LAYOUT	
[ID]	id	required		
[S]	cdata	implied		

## 21.7 SW-PARAM-RECORD-LAYOUT-CLASS

`[SW-PARAM-RECORD-LAYOUT-CLASS~]`—#PCDATA

Figure 206: DTD-diagram for SW-PARAM-RECORD-LAYOUT-CLASS

Child elements none

parent elements `<sw-param-record-layout>`

Table 205: Attributes for SW-PARAM-RECORD-LAYOUT-CLASS

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

## 21.8 SW-PARAM-RECORD-LAYOUT-DESC

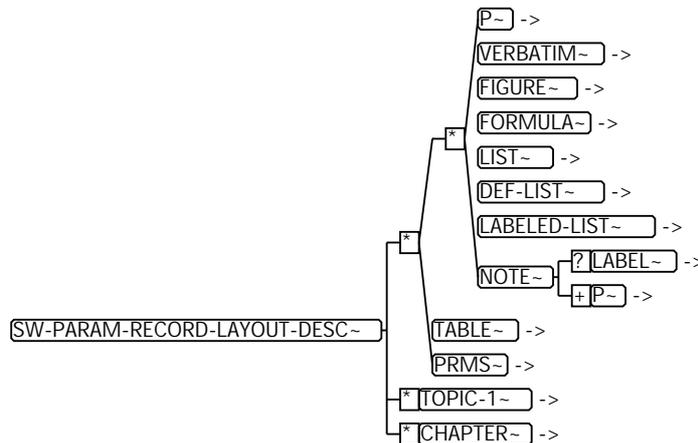


Figure 207: DTD-diagram for SW-PARAM-RECORD-LAYOUT-DESC

Child elements `<p>` `<verbatim>` `<figure>` `<formula>` `<list>` `<def-list>` `<labeled-list>` `<note>` `<table>` `<prms>` `<topic-1>` `<chapter>`

parent elements `<sw-param-record-layout>`

Table 206: Attributes for SW-PARAM-RECORD-LAYOUT-DESC

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

## 21.9 SW-PARAM-RECORD-LAYOUT-REF

`[SW-PARAM-RECORD-LAYOUT-REF~]`—#PCDATA

Figure 208: DTD-diagram for SW-PARAM-RECORD-LAYOUT-REF

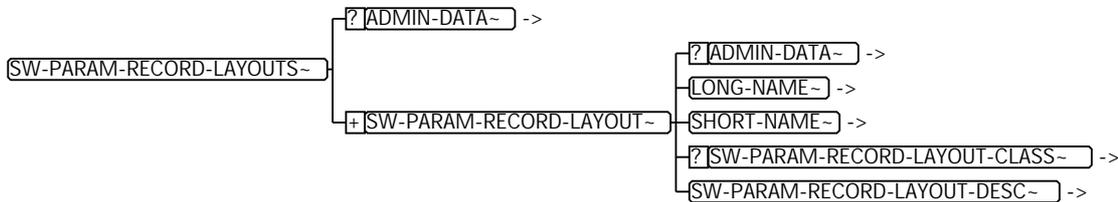
Child elements none

parent elements `<sw-param>`

**Table 207: Attributes for SW-PARAM-RECORD-LAYOUT-REF**

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND SW-PARAM-RECORD-LAYOUT	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		
[SW-PARAM-RECORD-LAYOUT]	idref	required		

## 21.10 SW-PARAM-RECORD-LAYOUTS



**Figure 209: DTD-diagram for SW-PARAM-RECORD-LAYOUTS**

Child elements `<admin-data>` `<sw-param-record-layout>`

parent elements `<sw-data-dictionary>`

**Table 208: Attributes for SW-PARAM-RECORD-LAYOUTS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 21.11 SW-PARAM-REF

`[SW-PARAM-REF~]—#PCDATA`

**Figure 210: DTD-diagram for SW-PARAM-REF**

Child elements none

parent elements `<sw-axis-grouped>` `<sw-param-content>` `<sw-param-refs>`

**Table 209: Attributes for SW-PARAM-REF**

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND SW-PARAM	
[HYTIME]	name	fixed	CLINK	
[OWNS]	nmtkgrp	implied	NOOWN	
[S]	cdata	implied		
[SW-PARAM]	idref	required		

## 21.12 SW-PARAM-REFS

`[SW-PARAM-REFS~]` `+ [SW-PARAM-REF~]` `#PCDATA`

Figure 211: DTD-diagram for SW-PARAM-REFS

Child elements `<sw-param-ref>`

parent elements `<sw-function-variant>`

Table 210: Attributes for SW-PARAM-REFS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 21.13 SW-PARAM-SINGLE-VALUE

`[SW-PARAM-SINGLE-VALUE~]` `[SW-PARAM-AXIS-VALUES~]` `[SW-COMPU-METHOD-REF]cdata`  
`? [SW-PARAM-NOEFFECT-VALUE~]` `#PCDATA`  
`? [SW-LIMITS~]` `->`  
`? [LABELS~]` `+ [LABEL~]` `->`

Figure 212: DTD-diagram for SW-PARAM-SINGLE-VALUE

Child elements `<sw-param-axis-values>`

parent elements `<sw-param>`

Table 211: Attributes for SW-PARAM-SINGLE-VALUE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 21.14 SW-PARAM-TARGET

`[SW-PARAM-TARGET~]` `[SW-VARIABLE-REF~]` `#PCDATA`

Figure 213: DTD-diagram for SW-PARAM-TARGET

Child elements `<sw-variable-ref>`

parent elements `<sw-param-group-axis>`

Table 212: Attributes for SW-PARAM-TARGET

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 21.15 SW-PARAM-TEXT

`[SW-PARAM-TEXT~]` - #PCDATA

Figure 214: DTD-diagram for SW-PARAM-TEXT

Child elements none

parent elements `<sw-param>`

Table 213: Attributes for SW-PARAM-TEXT

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 21.16 SW-PARAM-VALUE-BLOCK

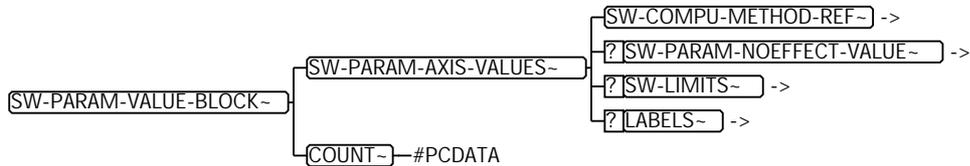


Figure 215: DTD-diagram for SW-PARAM-VALUE-BLOCK

Child elements `<sw-param-axis-values>` `<count>`

parent elements `<sw-param>`

Table 214: Attributes for SW-PARAM-VALUE-BLOCK

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 21.17 SW-PARAM-VALUES-ADR

`[SW-PARAM-VALUES-ADR~]` - `[V~]` - #PCDATA

Figure 216: DTD-diagram for SW-PARAM-VALUES-ADR

Child elements `<v>`

parent elements `<sw-param-content-v>` `<sw-param-content-x>` `<sw-param-content-y>`

Table 215: Attributes for SW-PARAM-VALUES-ADR

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 21.18 SW-PARAM-VALUES-CODED

`[SW-PARAM-VALUES-CODED~]` `[+V~]` `#PCDATA`

Figure 217: DTD-diagram for SW-PARAM-VALUES-CODED

Child elements `<v>`

parent elements `<sw-param-content-v>` `<sw-param-content-x>` `<sw-param-content-y>`

Table 216: Attributes for SW-PARAM-VALUES-CODED

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 21.19 SW-PARAM-VALUES-CODED-HEX

`[SW-PARAM-VALUES-CODED-HEX~]` `[+V~]` `#PCDATA`

Figure 218: DTD-diagram for SW-PARAM-VALUES-CODED-HEX

Child elements `<v>`

parent elements `<sw-param-content-v>` `<sw-param-content-x>` `<sw-param-content-y>`

Table 217: Attributes for SW-PARAM-VALUES-CODED-HEX

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 21.20 SW-PARAM-VALUES-GENERIC

`[SW-PARAM-VALUES-GENERIC~]` `[+V~]` `#PCDATA`

Figure 219: DTD-diagram for SW-PARAM-VALUES-GENERIC

Child elements `<v>`

parent elements `<sw-param-content-v>` `<sw-param-content-x>` `<sw-param-content-y>`

Table 218: Attributes for SW-PARAM-VALUES-GENERIC

Name	Type	Class	Value	Remark
[TYPE]	nmtoken	required		

## 22 SW-PARAM-VALUES-PHYS ... SW-VARIABLE-REF

### 22.1 SW-PARAM-VALUES-PHYS



Figure 220: DTD-diagram for SW-PARAM-VALUES-PHYS

Child elements <v>

parent elements <sw-param-content-v> <sw-param-content-x> <sw-param-content-y>

Table 219: Attributes for SW-PARAM-VALUES-PHYS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

### 22.2 SW-PARAMS

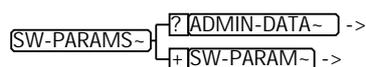


Figure 221: DTD-diagram for SW-PARAMS

Child elements <admin-data> <sw-param>

parent elements <sw-data-dictionary> <sw-param>

Table 220: Attributes for SW-PARAMS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.3 SW-PHYSIC-TYPE

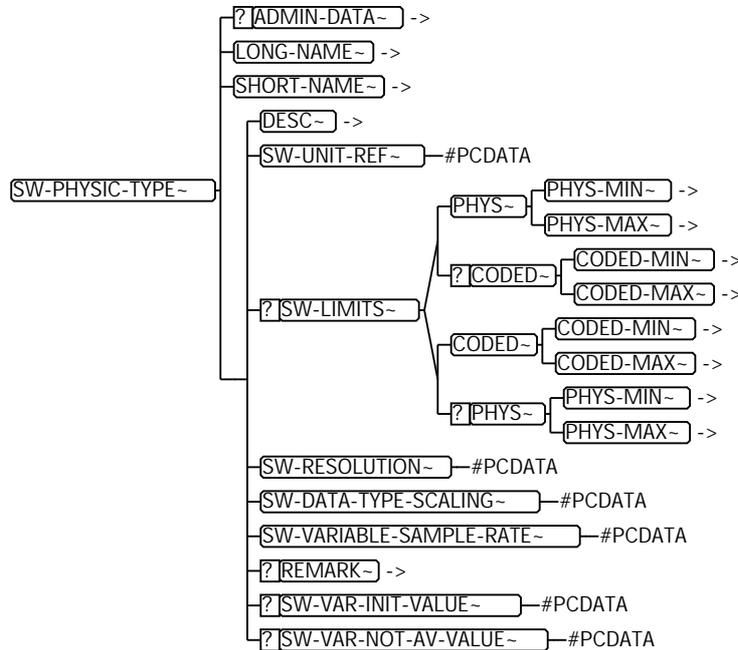


Figure 222: DTD-diagram for SW-PHYSIC-TYPE

Child elements `<admin-data>` `<long-name>` `<short-name>` `<desc>` `<sw-unit-ref>` `<sw-limits>` `<sw-resolution>` `<sw-data-type-scaling>` `<sw-variable-sample-rate>` `<remark>` `<sw-var-init-value>` `<sw-var-not-av-value>`

parent elements `<sw-physic-types>`

Table 221: Attributes for SW-PHYSIC-TYPE

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	SW-PHYSIC-TYPE	
[ID]	id	required		
[S]	cdata	implied		

## 22.4 SW-PHYSIC-TYPE-1

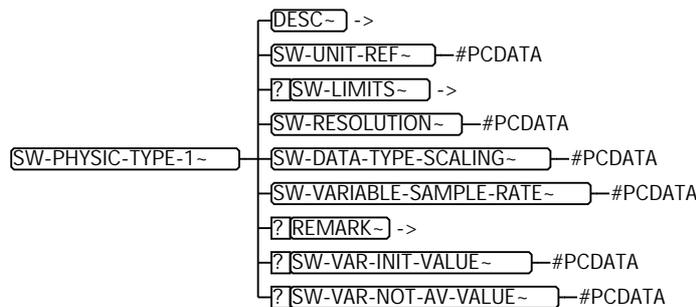


Figure 223: DTD-diagram for SW-PHYSIC-TYPE-1

Child elements **<desc>** **<sw-unit-ref>** **<sw-limits>** **<sw-resolution>** **<sw-data-type-scaling>** **<sw-variable-sample-rate>** **<remark>** **<sw-var-init-value>** **<sw-var-not-av-value>**

parent elements **<sw-variable>**

**Table 222: Attributes for SW-PHYSIC-TYPE-1**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.5 SW-PHYSIC-TYPE-REF

**[SW-PHYSIC-TYPE-REF~]** → #PCDATA

**Figure 224: DTD-diagram for SW-PHYSIC-TYPE-REF**

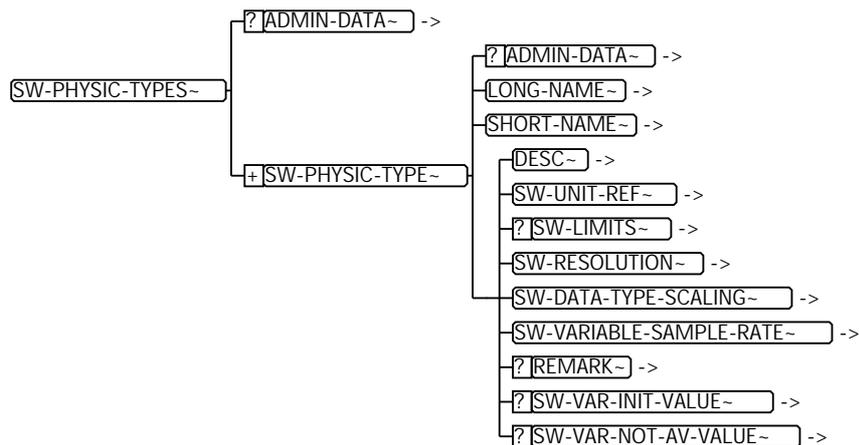
Child elements none

parent elements **<sw-variable>**

**Table 223: Attributes for SW-PHYSIC-TYPE-REF**

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND SW-PHYSIC-TYPE	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		
[SW-PHYSIC-TYPE]	idref	required		

## 22.6 SW-PHYSIC-TYPES



**Figure 225: DTD-diagram for SW-PHYSIC-TYPES**

Child elements **<admin-data>** **<sw-physic-type>**

parent elements **<sw-data-dictionary>**

**Table 224: Attributes for SW-PHYSIC-TYPES**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.7 SW-RESOLUTION

`[SW-RESOLUTION~]` - #PCDATA

**Figure 226: DTD-diagram for SW-RESOLUTION**

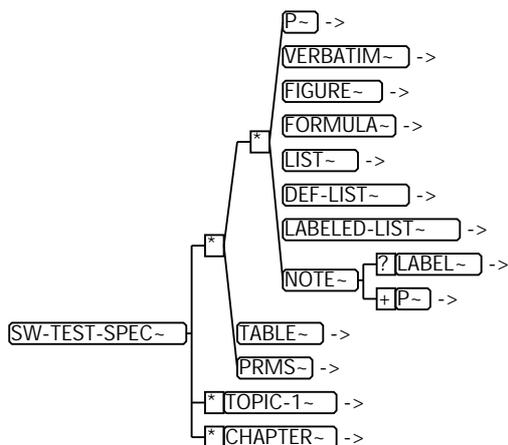
Child elements none

parent elements `<sw-physic-type>` `<sw-physic-type-1>`

**Table 225: Attributes for SW-RESOLUTION**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.8 SW-TEST-SPEC



**Figure 227: DTD-diagram for SW-TEST-SPEC**

Child elements `<p>` `<verbatim>` `<figure>` `<formula>` `<list>` `<def-list>` `<labeled-list>` `<note>`  
`<table>` `<prms>` `<topic-1>` `<chapter>`

parent elements `<sw-function-variant>`

**Table 226: Attributes for SW-TEST-SPEC**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.9 SW-UNIT

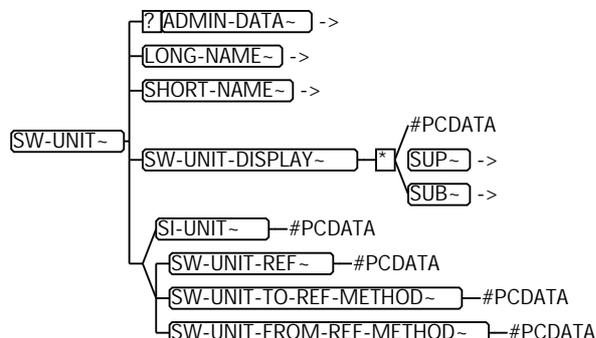


Figure 228: DTD-diagram for SW-UNIT

Child elements **<admin-data>** **<long-name>** **<short-name>** **<sw-unit-display>** **<si-unit>**  
**<sw-unit-ref>** **<sw-unit-to-ref-method>** **<sw-unit-from-ref-method>**

parent elements **<sw-units>**

Table 227: Attributes for SW-UNIT

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	SW-UNIT	
[ID]	id	required		
[S]	cdata	implied		

## 22.10 SW-UNIT-DISPLAY

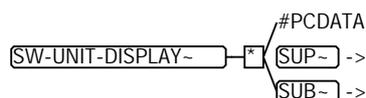


Figure 229: DTD-diagram for SW-UNIT-DISPLAY

Child elements **<sup>** **<sub>**

parent elements **<sw-unit>**

Table 228: Attributes for SW-UNIT-DISPLAY

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.11 SW-UNIT-FROM-REF-METHOD



Figure 230: DTD-diagram for SW-UNIT-FROM-REF-METHOD

Child elements none



parent elements **<sw-unit>**

**Table 229: Attributes for SW-UNIT-FROM-REF-METHOD**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.12 SW-UNIT-REF

**SW-UNIT-REF~** —#PCDATA

**Figure 231: DTD-diagram for SW-UNIT-REF**

Child elements none

parent elements **<sw-compu-method>** **<sw-param-content-v>** **<sw-param-content-x>** **<sw-param-content-y>** **<sw-physic-type>** **<sw-physic-type-1>** **<sw-unit>**

**Table 230: Attributes for SW-UNIT-REF**

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND SW-UNIT	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		
[SW-UNIT]	idref	required		

## 22.13 SW-UNIT-TO-REF-METHOD

**SW-UNIT-TO-REF-METHOD~** —#PCDATA

**Figure 232: DTD-diagram for SW-UNIT-TO-REF-METHOD**

Child elements none

parent elements **<sw-unit>**

**Table 231: Attributes for SW-UNIT-TO-REF-METHOD**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.14 SW-UNITS

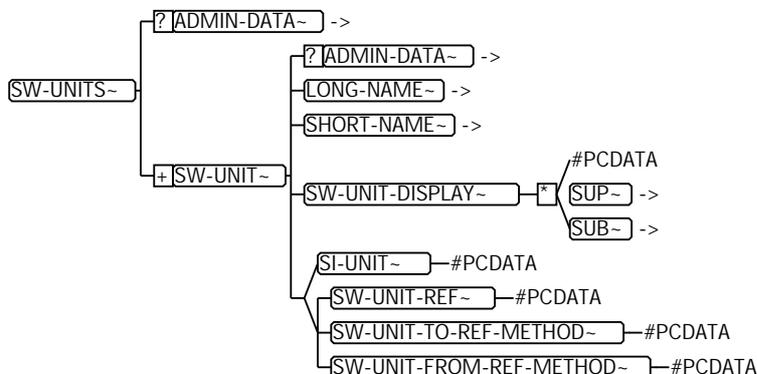


Figure 233: DTD-diagram for SW-UNITS

Child elements **<admin-data>** **<sw-unit>**

parent elements **<sw-data-dictionary>**

Table 232: Attributes for SW-UNITS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.15 SW-VAR-INIT-VALUE



Figure 234: DTD-diagram for SW-VAR-INIT-VALUE

Child elements none

parent elements **<sw-physic-type>** **<sw-physic-type-1>** **<sw-variable-implementation>**

Table 233: Attributes for SW-VAR-INIT-VALUE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.16 SW-VAR-NOT-AV-VALUE



Figure 235: DTD-diagram for SW-VAR-NOT-AV-VALUE

Child elements none

parent elements **<sw-physic-type>** **<sw-physic-type-1>** **<sw-variable-implementation>**

Table 234: Attributes for SW-VAR-NOT-AV-VALUE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.17 SW-VARIABLE

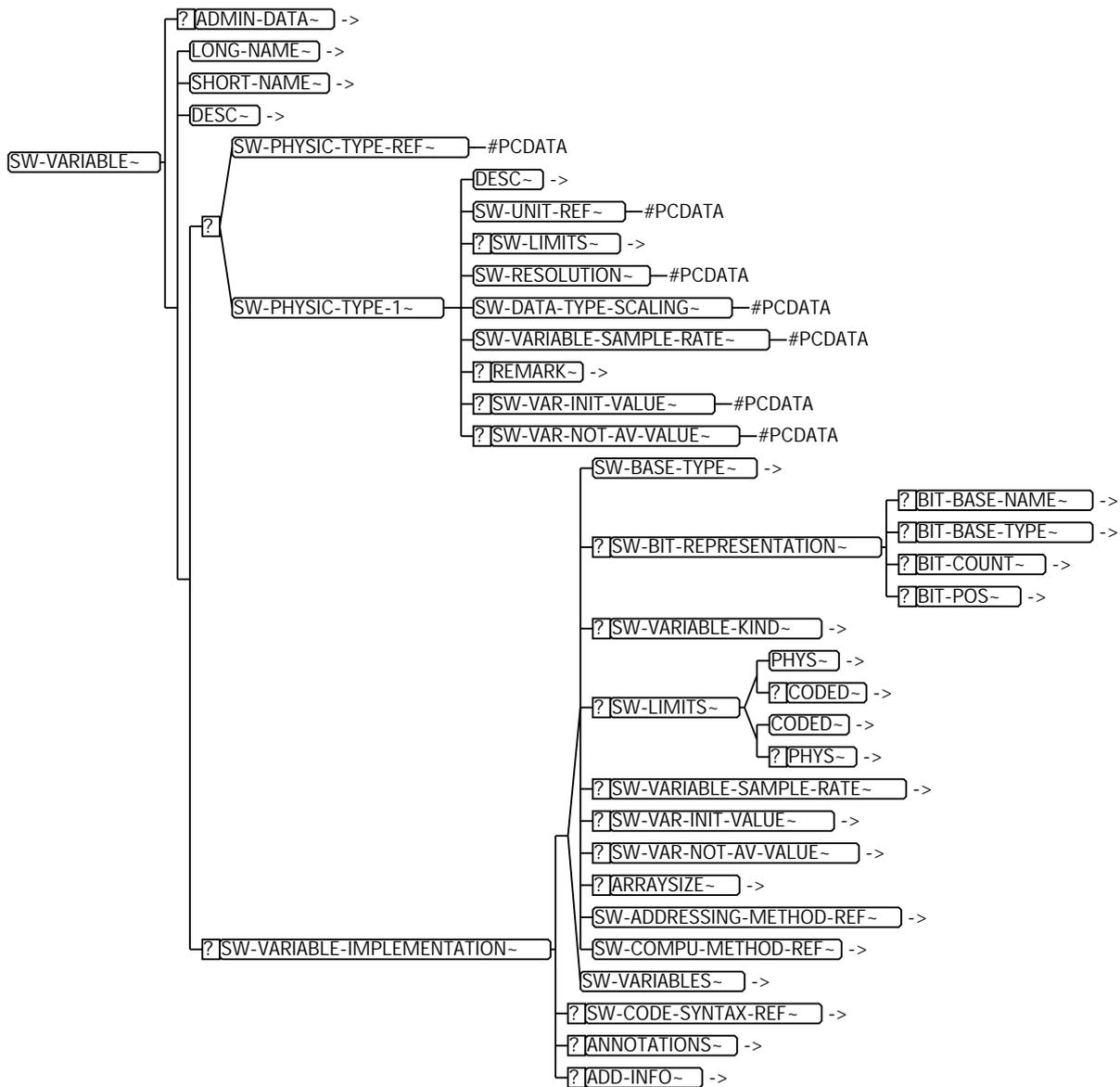


Figure 236: DTD-diagram for SW-VARIABLE

Child elements `<admin-data>` `<long-name>` `<short-name>` `<desc>` `<sw-physic-type-ref>`  
`<sw-physic-type-1>` `<sw-variable-implementation>`

parent elements `<sw-variables>`

Table 235: Attributes for SW-VARIABLE

Name	Type	Class	Value	Remark
[CALIBRATION]	nmtkgrp	default	CALIBRATION NOT-IN-MEMORY CALIBRATION NO- CALIBRATION	
[F-ID-CLASS]	name	fixed	SW-VARIABLE	
[F-NAMESPACE]	names	fixed	SW-VARIABLE	
[ID]	id	required		
[S]	cdata	implied		

## 22.18

## SW-VARIABLE-IMPLEMENTATION

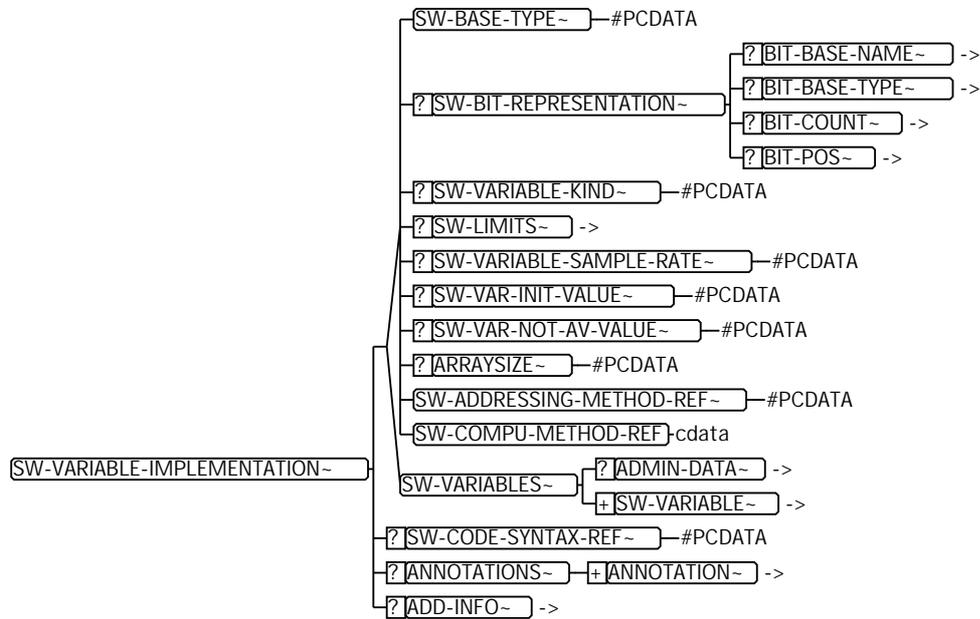


Figure 237: DTD-diagram for SW-VARIABLE-IMPLEMENTATION

Child elements `<sw-base-type>` `<sw-bit-representation>` `<sw-variable-kind>` `<sw-limits>`  
`<sw-variable-sample-rate>` `<sw-var-init-value>` `<sw-var-not-av-value>`  
`<arraysize>` `<sw-addressing-method-ref>` `<sw-compu-method-ref>` `<sw-variables>` `<sw-code-syntax-ref>` `<annotations>` `<add-info>`

parent elements `<sw-variable>`

Table 236: Attributes for SW-VARIABLE-IMPLEMENTATION

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.19 SW-VARIABLE-KIND

`SW-VARIABLE-KIND~` —#PCDATA

Figure 238: DTD-diagram for SW-VARIABLE-KIND

Child elements none

parent elements `<sw-variable-implementation>`

Table 237: Attributes for SW-VARIABLE-KIND

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 22.20 SW-VARIABLE-REF

`SW-VARIABLE-REF~` —#PCDATA

Figure 239: DTD-diagram for SW-VARIABLE-REF

Child elements none

parent elements `<sw-axis-individual>` `<sw-axis-shift-offset>` `<sw-function-modelonly-variables>` `<sw-param-target>` `<sw-variables-read>` `<sw-variables-readwrite>` `<sw-variables-write>`

Table 238: Attributes for SW-VARIABLE-REF

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND SW-VARIABLE	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		
[SW-VARIABLE]	idref	required		

## 23 SW-VARIABLE-SAMPLE-RATE ... SYSTEM-OVERVIEW

### 23.1 SW-VARIABLE-SAMPLE-RATE

`[SW-VARIABLE-SAMPLE-RATE~]` — #PCDATA

Figure 240: DTD-diagram for SW-VARIABLE-SAMPLE-RATE

Child elements none

parent elements `<sw-physic-type>` `<sw-physic-type-1>` `<sw-variable-implementation>`

Table 239: Attributes for SW-VARIABLE-SAMPLE-RATE

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

### 23.2 SW-VARIABLES

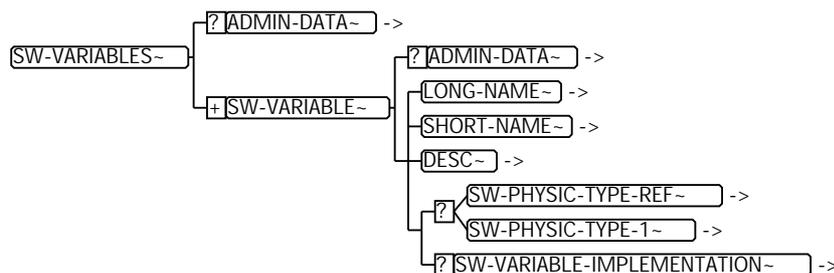


Figure 241: DTD-diagram for SW-VARIABLES

Child elements `<admin-data>` `<sw-variable>`

parent elements `<sw-data-dictionary>` `<sw-variable-implementation>`

Table 240: Attributes for SW-VARIABLES

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

### 23.3 SW-VARIABLES-READ

`[SW-VARIABLES-READ~]` — + `[SW-VARIABLE-REF~]` — #PCDATA

Figure 242: DTD-diagram for SW-VARIABLES-READ

Child elements `<sw-variable-ref>`

parent elements `<sw-function-export-variables>` `<sw-function-import-variables>`

**Table 241: Attributes for SW-VARIABLES-READ**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 23.4 SW-VARIABLES-READWRITE

`SW-VARIABLES-READWRITE~` `+SW-VARIABLE-REF~` `#PCDATA`

**Figure 243: DTD-diagram for SW-VARIABLES-READWRITE**

Child elements `<sw-variable-ref>`

parent elements `<sw-function-export-variables>` `<sw-function-import-variables>` `<sw-function-local-variables>`

**Table 242: Attributes for SW-VARIABLES-READWRITE**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 23.5 SW-VARIABLES-WRITE

`SW-VARIABLES-WRITE~` `+SW-VARIABLE-REF~` `#PCDATA`

**Figure 244: DTD-diagram for SW-VARIABLES-WRITE**

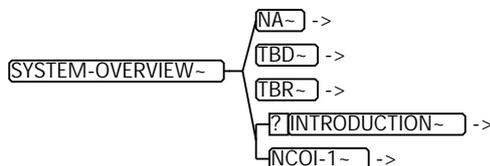
Child elements `<sw-variable-ref>`

parent elements `<sw-function-export-variables>` `<sw-function-import-variables>`

**Table 243: Attributes for SW-VARIABLES-WRITE**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 23.6 SYSTEM-OVERVIEW



**Figure 245: DTD-diagram for SYSTEM-OVERVIEW**

Child elements `<na>` `<tbd>` `<tbr>` `<introduction>` `<ncoi-1>`

parent elements `<general-project-data>`



**Table 244: Attributes for SYSTEM-OVERVIEW**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 24 TABLE ... TYP

### 24.1 TABLE

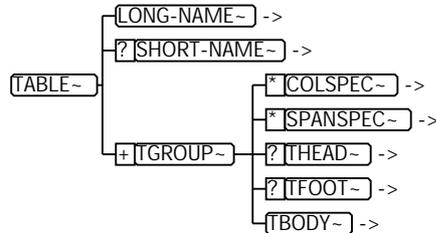


Figure 246: DTD-diagram for TABLE

Child elements **<long-name>** **<short-name>** **<tgroup>**

parent elements **<add-info>** **<chapter>** **<introduction>** **<ncoi-1>** **<sw-addressing-method-desc>** **<sw-application-notes>** **<sw-carb-doc>** **<sw-code-syntax-desc>** **<sw-function-desc>** **<sw-maintenance-notes>** **<sw-param-record-layout-desc>** **<sw-test-spec>** **<topic-1>** **<topic-2>**

Table 245: Attributes for TABLE

Name	Type	Class	Value	Remark
[COLSEP]	number	implied		
[F-ID-CLASS]	name	fixed	TABLE	
[FLOAT]	nmtkgrp	implied	FLOAT NO-FLOAT	
[FRAME]	nmtkgrp	implied	TOP BOTTOM TOP-BOT ALL SIDES NONE	
[HELP-ENTRY]	cdata	implied		
[ID]	id	required		
[ORIENT]	nmtkgrp	implied	PORT LAND	
[PGWIDE]	number	implied		
[ROWSEP]	number	implied		
[S]	cdata	implied		
[SHORTENTRY]	number	implied		
[TABSTYLE]	nmtoken	implied		
[TOCENTRY]	number	default	1	

## 24.2 TBD

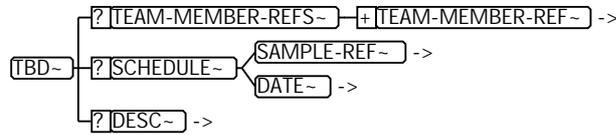


Figure 247: DTD-diagram for TBD

Child elements <team-member-refs> <schedule> <desc>

parent elements <acceptance-cond> <add-spec> <demarcation-other-projects> <dir-hand-over-doc-data> <general-project-data> <integration-capability> <objectives> <parallel-designs> <project-schedule> <protocols> <purchasing-cond> <reason-order> <sample-spec> <sw-data-dictionary-spec> <sw-function-spec> <sw-glossary> <sw-param-contents-spec> <system-overview> <variant-spec>

Table 246: Attributes for TBD

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 24.3 TBODY



Figure 248: DTD-diagram for TBODY

Child elements <row>

parent elements <tgroup>

Table 247: Attributes for TBODY

Name	Type	Class	Value	Remark
[S]	cdata	implied		
[VALIGN]	nmtkgrp	default	TOP TOP MIDDLE BOT- TOM	

## 24.4 TBR

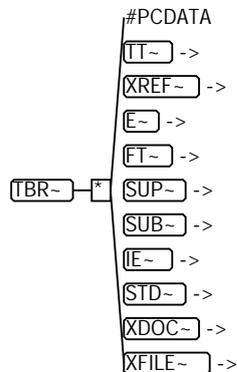


Figure 249: DTD-diagram for TBR

Child elements `<tt>` `<xref>` `<e>` `<ft>` `<sup>` `<sub>` `<ie>` `<std>` `<xdoc>` `<xfile>`

parent elements `<acceptance-cond>` `<add-spec>` `<demarcation-other-projects>` `<dir-hand-over-doc-data>` `<general-project-data>` `<integration-capability>` `<objectives>` `<parallel-designs>` `<project-schedule>` `<protocols>` `<purchasing-cond>` `<reason-order>` `<sample-spec>` `<sw-data-dictionary-spec>` `<sw-function-spec>` `<sw-glossary>` `<sw-param-contents-spec>` `<system-overview>` `<variant-spec>`

Table 248: Attributes for TBR

Name	Type	Class	Value	Remark
[S]	CDATA	implied		

## 24.5 TEAM-MEMBER

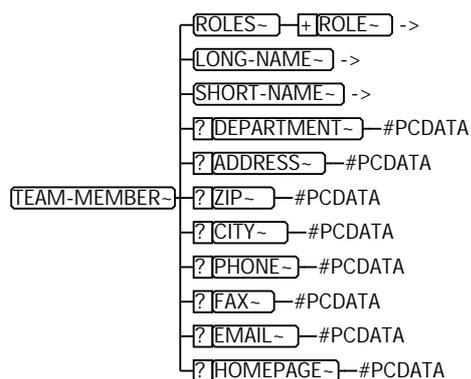


Figure 250: DTD-diagram for TEAM-MEMBER

Child elements `<roles>` `<long-name>` `<short-name>` `<department>` `<address>` `<zip>` `<city>` `<phone>` `<fax>` `<email>` `<homepage>`

parent elements `<team-members>`

**Table 249: Attributes for TEAM-MEMBER**

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	TEAM-MEMBER	
[ID]	id	required		
[S]	cdata	implied		

## 24.6 TEAM-MEMBER-REF

`[TEAM-MEMBER-REF-]`—#PCDATA

**Figure 251: DTD-diagram for TEAM-MEMBER-REF**

Child elements none

parent elements `<doc-revision>` `<team-member-refs>`

**Table 250: Attributes for TEAM-MEMBER-REF**

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND TEAM-MEMBER	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		
[TEAM-MEMBER]	idref	required		

## 24.7 TEAM-MEMBER-REFS

`[TEAM-MEMBER-REFS-]`+`[TEAM-MEMBER-REF-]`—#PCDATA

**Figure 252: DTD-diagram for TEAM-MEMBER-REFS**

Child elements `<team-member-ref>`

parent elements `<tbd>`

**Table 251: Attributes for TEAM-MEMBER-REFS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 24.8 TEAM-MEMBERS

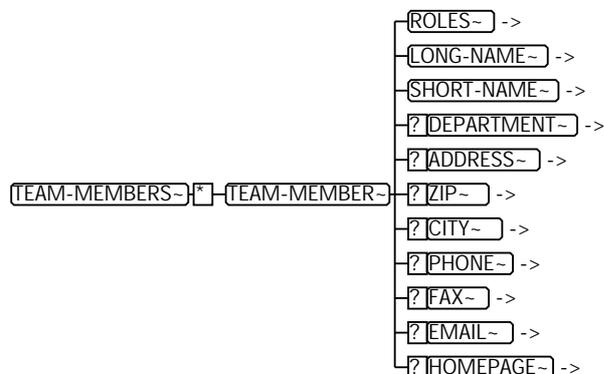


Figure 253: DTD-diagram for TEAM-MEMBERS

Child elements <team-member>

parent elements <company>

Table 252: Attributes for TEAM-MEMBERS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 24.9 TEX-MATH



Figure 254: DTD-diagram for TEX-MATH

Child elements none

parent elements <formula>

Table 253: Attributes for TEX-MATH

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 24.10 TEXT

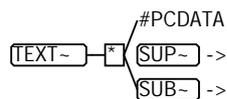


Figure 255: DTD-diagram for TEXT

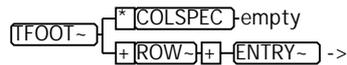
Child elements <sup> <sub>

parent elements <prm-char>

**Table 254: Attributes for TEXT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 24.11 TFOOT



**Figure 256: DTD-diagram for TFOOT**

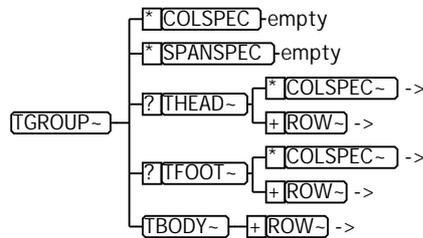
Child elements `<colspec>` `<row>`

parent elements `<tgroup>`

**Table 255: Attributes for TFOOT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		
[VALIGN]	nmtkgrp	default	TOP TOP MIDDLE BOT- TOM	

## 24.12 TGROUP



**Figure 257: DTD-diagram for TGROUP**

Child elements `<colspec>` `<spanspec>` `<thead>` `<tfoot>` `<tbody>`

parent elements `<table>`

**Table 256: Attributes for TGROUP**

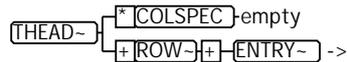
Name	Type	Class	Value	Remark
[ALIGN]	nmtkgrp	default	LEFT LEFT RIGHT CEN- TER JUSTIFY CHAR	
[CHAR]	cdata	default		
[CHAROFF]	nutoken	default	50	
[COLS]	number	required		
[COLSEP]	number	implied		
[ROWSEP]	number	implied		

**Table 256 (Cont.): Attributes for TGROUP**

Name	Type	Class	Value	Remark
[S]	cdata	implied		
[TGROUPSTYLE]	nmtoken	implied		

## 24.13

### THEAD



**Figure 258: DTD-diagram for THEAD**

Child elements `<colspec>` `<row>`

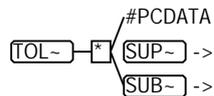
parent elements `<tgroup>`

**Table 257: Attributes for THEAD**

Name	Type	Class	Value	Remark
[S]	cdata	implied		
[VALIGN]	nmtkgrp	default	BOTTOM TOP MIDDLE BOT- TOM	

## 24.14

### TOL



**Figure 259: DTD-diagram for TOL**

Child elements `<sup>` `<sub>`

parent elements `<prm-char>`

**Table 258: Attributes for TOL**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 24.15 TOPIC-1

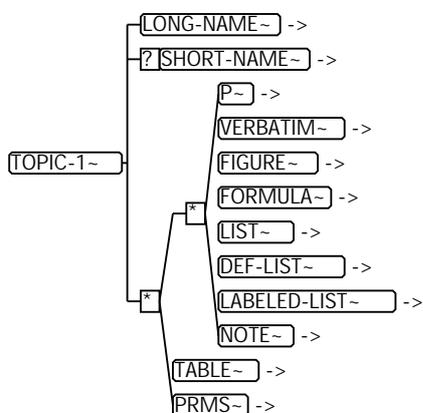


Figure 260: DTD-diagram for TOPIC-1

Child elements **<long-name>** **<short-name>** **<p>** **<verbatim>** **<figure>** **<formula>** **<list>** **<def-list>** **<labeled-list>** **<note>** **<table>** **<prms>**

parent elements **<add-info>** **<chapter>** **<ncoi-1>** **<sw-addressing-method-desc>** **<sw-application-notes>** **<sw-carb-doc>** **<sw-code-syntax-desc>** **<sw-function-desc>** **<sw-maintenance-notes>** **<sw-param-record-layout-desc>** **<sw-test-spec>**

Table 259: Attributes for TOPIC-1

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	TOPIC	
[HELP-ENTRY]	cdata	implied		
[ID]	id	required		
[S]	cdata	implied		

## 24.16 TOPIC-2

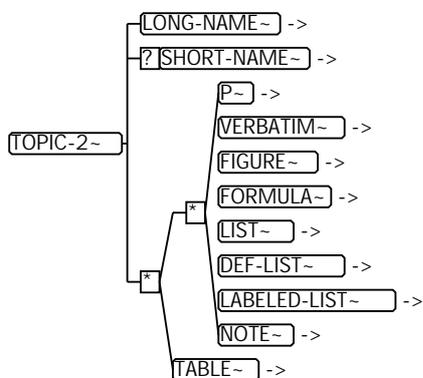


Figure 261: DTD-diagram for TOPIC-2

Child elements **<long-name>** **<short-name>** **<p>** **<verbatim>** **<figure>** **<formula>** **<list>** **<def-list>** **<labeled-list>** **<note>** **<table>**

parent elements **<introduction>**

**Table 260: Attributes for TOPIC-2**

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	TOPIC	
[HELP-ENTRY]	cdata	implied		
[ID]	id	required		
[S]	cdata	implied		

## 24.17

## TT

 #PCDATA

**Figure 262: DTD-diagram for TT**

Child elements none

parent elements **<change>** **<desc>** **<indent-sample>** **<item-label>** **<long-name>** **<long-name-1>** **<p>** **<reason>** **<tbr>**

**Table 261: Attributes for TT**

Name	Type	Class	Value	Remark
[S]	cdata	implied		
[TYPE]	nmtkgrp	required	SGMLTAG SGML-ATTRIBUTE TOOL PRODUCT VARIABLE STATE PRM MATERIAL CONTROL-ELEMENT CODE ORGANISATION OTHER	
[USER-DEFINED-TYPE]	cdata	implied		

## 24.18

## TYP

 \*  -> #PCDATA  
 ->

**Figure 263: DTD-diagram for TYP**

Child elements **<sup>** **<sub>**

parent elements **<prm-char>**

**Table 262: Attributes for TYP**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 25 UNIT ... USED-LANGUAGES

### 25.1 UNIT

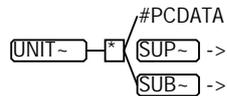


Figure 264: DTD-diagram for UNIT

Child elements **<sup>** **<sub>**

parent elements **<prm-char>**

Table 263: Attributes for UNIT

Name	Type	Class	Value	Remark
[S]	cdata	implied		

### 25.2 USED-LANGUAGES



Figure 265: DTD-diagram for USED-LANGUAGES

Child elements none

parent elements **<admin-data>**

Table 264: Attributes for USED-LANGUAGES

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 26 V ... VERBATIM

### 26.1 V

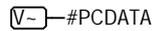
#PCDATA

Figure 266: DTD-diagram for V

Child elements none

parent elements <sw-param-values-adr> <sw-param-values-coded> <sw-param-values-coded-hex> <sw-param-values-generic> <sw-param-values-phys>

Table 265: Attributes for V

Name	Type	Class	Value	Remark
[S]	cdata	implied		

### 26.2 VALUE

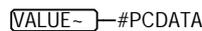
#PCDATA

Figure 267: DTD-diagram for VALUE

Child elements none

parent elements <variant-char-value>

Table 266: Attributes for VALUE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

### 26.3 VARIANT-CHAR

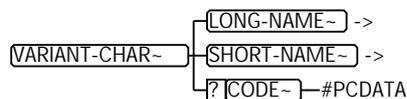


Figure 268: DTD-diagram for VARIANT-CHAR

Child elements <long-name> <short-name> <code>

parent elements <variant-chars>

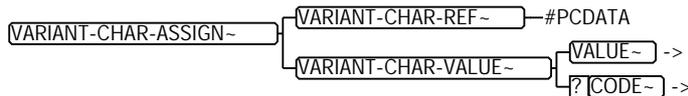
Table 267: Attributes for VARIANT-CHAR

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	VARIANT-CHAR	
[ID]	id	required		
[S]	cdata	implied		

**Table 267 (Cont.): Attributes for VARIANT-CHAR**

Name	Type	Class	Value	Remark
[TYPE]	nmtkgrp	required	NEW-PART-NUMBER NO-NEW-PART-NUMBER	

## 26.4 VARIANT-CHAR-ASSIGN



**Figure 269: DTD-diagram for VARIANT-CHAR-ASSIGN**

Child elements <variant-char-ref> <variant-char-value>

parent elements <variant-char-assigns>

**Table 268: Attributes for VARIANT-CHAR-ASSIGN**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 26.5 VARIANT-CHAR-ASSIGNS



**Figure 270: DTD-diagram for VARIANT-CHAR-ASSIGNS**

Child elements <variant-char-assign>

parent elements <variant-def>

**Table 269: Attributes for VARIANT-CHAR-ASSIGNS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 26.6 VARIANT-CHAR-REF



**Figure 271: DTD-diagram for VARIANT-CHAR-REF**

Child elements none

parent elements <variant-char-assign>

Table 270: Attributes for VARIANT-CHAR-REF

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND VARIANT-CHAR	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		
[VARIANT-CHAR]	idref	required		

## 26.7 VARIANT-CHAR-VALUE

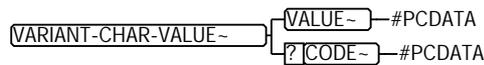


Figure 272: DTD-diagram for VARIANT-CHAR-VALUE

Child elements <value> <code>

parent elements <variant-char-assign>

Table 271: Attributes for VARIANT-CHAR-VALUE

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 26.8 VARIANT-CHARS



Figure 273: DTD-diagram for VARIANT-CHARS

Child elements <variant-char>

parent elements <variant-spec>

Table 272: Attributes for VARIANT-CHARS

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 26.9 VARIANT-DEF



Figure 274: DTD-diagram for VARIANT-DEF

Child elements <long-name> <short-name> <code> <variant-char-assigns>



parent elements **<variant-defs>**

**Table 273: Attributes for VARIANT-DEF**

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	VARIANT-DEF	
[ID]	id	required		
[S]	cdata	implied		

## 26.10 VARIANT-DEF-REF

**VARIANT-DEF-REF~** — #PCDATA

**Figure 275: DTD-diagram for VARIANT-DEF-REF**

Child elements none

parent elements **<variant-def-refs>**

**Table 274: Attributes for VARIANT-DEF-REF**

Name	Type	Class	Value	Remark
[HYNAMES]	names	fixed	LINKEND VARIANT-DEF	
[HYTIME]	name	fixed	CLINK	
[S]	cdata	implied		
[VARIANT-DEF]	idref	required		

## 26.11 VARIANT-DEF-REFS

**VARIANT-DEF-REFS~** — + **VARIANT-DEF-REF~** — #PCDATA

**Figure 276: DTD-diagram for VARIANT-DEF-REFS**

Child elements **<variant-def-ref>**

parent elements **<sw-function-variant>**

**Table 275: Attributes for VARIANT-DEF-REFS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 26.12 VARIANT-DEFS

**VARIANT-DEFS~** — + **VARIANT-DEF~** —

- LONG-NAME~** ->
- SHORT-NAME~** ->
- CODE~** ->
- VARIANT-CHAR-ASSIGNS~** ->

**Figure 277: DTD-diagram for VARIANT-DEFS**

Child elements **<variant-def>**

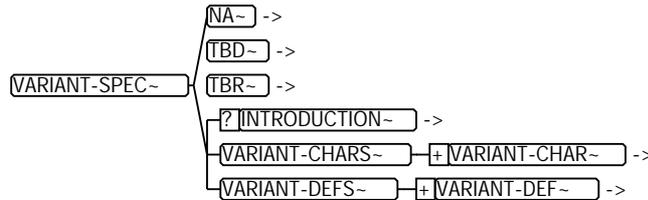


parent elements **<variant-spec>**

**Table 276: Attributes for VARIANT-DEFS**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 26.13 VARIANT-SPEC



**Figure 278: DTD-diagram for VARIANT-SPEC**

Child elements **<na> <tbd> <tbr> <introduction> <variant-chars> <variant-defs>**

parent elements **<general-project-data>**

**Table 277: Attributes for VARIANT-SPEC**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## 26.14 VERBATIM



**Figure 279: DTD-diagram for VERBATIM**

Child elements none

parent elements **<add-info> <annotation-text> <chapter> <entry> <figure> <formula> <introduction> <item> <labeled-item> <ncoi-1> <remark> <sw-addressing-method-desc> <sw-application-notes> <sw-carb-doc> <sw-code-syntax-desc> <sw-function-desc> <sw-maintenance-notes> <sw-param-record-layout-desc> <sw-test-spec> <topic-1> <topic-2>**

**Table 278: Attributes for VERBATIM**

Name	Type	Class	Value	Remark
[ALLOW-BREAK]	number	default	1	
[FLOAT]	nmtkgrp	implied	FLOAT NO-FLOAT	
[HELP-ENTRY]	cdata	implied		
[S]	cdata	implied		

## 27 XDOC ... XREF

### 27.1 XDOC

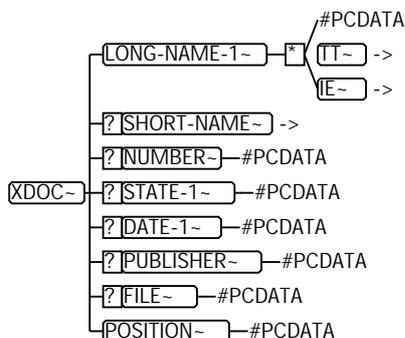


Figure 280: DTD-diagram for XDOC

Child elements **<long-name-1>** **<short-name>** **<number>** **<state-1>** **<date-1>** **<publisher>**  
**<file>** **<position>**

parent elements **<p>** **<tbr>**

Table 279: Attributes for XDOC

Name	Type	Class	Value	Remark
[F-CHILD-TYPE]	cdata	fixed	date-1:date	
[F-ID-CLASS]	name	fixed	XDOC	
[ID]	id	required		
[S]	cdata	implied		

### 27.2 XFILE

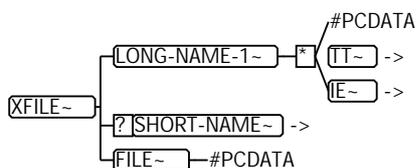


Figure 281: DTD-diagram for XFILE

Child elements **<long-name-1>** **<short-name>** **<file>**

parent elements **<p>** **<tbr>**

Table 280: Attributes for XFILE

Name	Type	Class	Value	Remark
[F-ID-CLASS]	name	fixed	XFILE	
[ID]	id	required		
[S]	cdata	implied		



## 27.3 XREF

`<XREF~>#PCDATA`

Figure 282: DTD-diagram for XREF

Child elements none

parent elements `<change>` `<desc>` `<indent-sample>` `<item-label>` `<p>` `<reason>` `<tbr>`

Table 281: Attributes for XREF

Name	Type	Class	Value	Remark
[EXT-ID-CLASS]	cdata	implied		
[HYNAMES]	names	fixed	LINKEND ID-REF	
[HYTIME]	name	fixed	CLINK	
[ID-CLASS]	nmtkgrp	required	CHAPTER COMPANY DEF-ITEM FIGURE FORMULA PRM SAMPLE STD SW-FUNCTION SW- UNIT SW-PHYSIC- TYPE SW-VARIABLE SW-PARAM SW- COMPU-METHOD SW-ADDRESSING- METHOD SW- PARAM-RECORD- LAYOUT SW-CODE- SYNTAX TABLE TEAM-MEMBER TOPIC VARIANT- CHAR VARIANT-DEF XDOC XFILE EXTER- NAL	
[ID-REF]	idref	required		
[S]	cdata	implied		



## 28 ZIP ... ZIP

### 28.1 ZIP

#PCDATA

**Figure 283: DTD-diagram for ZIP**

Child elements none

parent elements <team-member>

**Table 282: Attributes for ZIP**

Name	Type	Class	Value	Remark
[S]	cdata	implied		

## Documentadministration

**Table : team members**

Name	Company
Dipl.-Ing.(FH) Uwe Bless	MSR-MEDOC
Dipl.-Inform. Helmut Gengenbach	MSR-MEDOC
Dipl. Ing. Eckard Jakobi	MSR-MEDOC
Dipl.-Ing. Herbert Klein	MSR-MEDOC
Dipl. Ing. Oliver Marcks	MSR-MEDOC
Dipl.-Inform. Peter Rauleder	MSR-MEDOC
Dipl.-Ing. Martin Trinschek	MSR-MEDOC
Dipl.-Ing. Bernhard Weichel	MSR-MEDOC

**Table : version overview**

Date	Publisher
19.2.1999	Dipl.-Ing. Bernhard Weichel

**Table : modifications**

Change	Related to
initial revision Reason:	Content

**Table : modifications included**

Date	Chapter	Change	Related to
Nr. 1, 19.2.1999	Gesamt	initial revision Reason:	Content