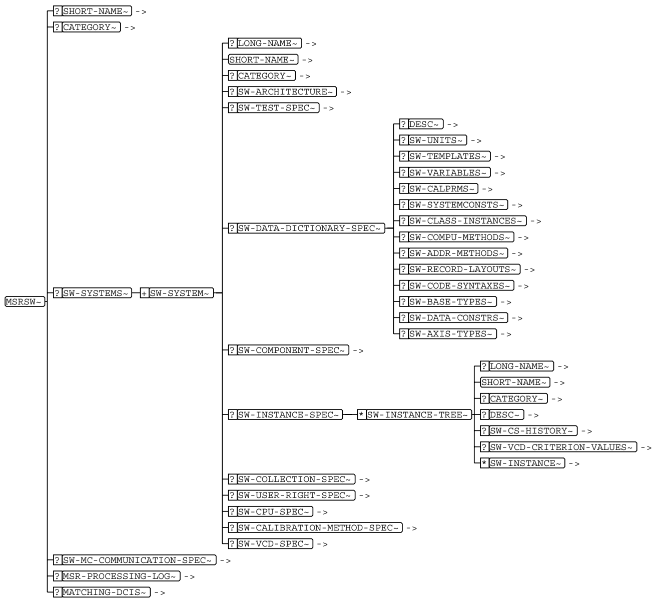
# ASAM-MCD-2MC V2.0 / MSRSW V2.2.0

# Introduction for ASAM-MCD-2MC audience

## DTD final draft




**ASAM-MCD and MSRSW joint taskforce**, Bernhard Weichel, Robert Bosch GmbH

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# Abstract

This document gives a first introduction in *MSRSW.DTD*, especially for the ASAM-MCD community. Therefore it

- contains basic principles of SGML and MSR
- illustrates primarily the ASAM-MCD-2MC usecase
- does not cover all aspects to its full extent
- gives an outlook about the additional new possibilities

The document was edited for preparation of the ASAM-MCD meeting scheduled for 5.4.2001. We apologize for incompleteness resp. for some language mix in some figures as well as for any errors ...

**Any recommendations for improvements and hints to bugs are very welcome by ”bernhard.weichel@de.bosch.com”.**

***Caution:***
This document was originally prepared for prerelease 4 of the ASAM-MCD-2MC DTDS. It was upgraded to the the final draft. Therefore some examples and figures still may reflect prerelease 4.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# Table of Contents



All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# List of Figures



All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# Introduction

*Companies*

## ASAM-MCD and MSRSW joint taskforce [ASAM-MSR]

| **Name**<br>**Roles** | **Departement** | **Address** | **Contact** |
|---|---|---|---|
| Lutz Berentroth<br>ASAM-Representative | Siemens AT | | |
| Uwe Bless | Daimer-Chrysler | | |
| William Debs, | GM Power-train | | |
| Michael Epping<br>MSR-Representative | Hella | | |
| Helmut Gengenbach<br>MSR-Representative | Robert Bosch GmbH | | |
| Ute Gerlach<br>ASAM-Representative | ETAS GmbH | | |
| Eckard Jakobi<br>MSR-Representative | Siemens AT | | |
| Ola Josefsson<br>ASAM-Representative | Volvo | | |
| Herbert Klein<br>MSR-Representative, DTD architect, TZKOM GmbH | XI-works | | |
| H. Kunz<br>ASAM-Representative | VDO | | |
| Christian Larsson<br>ASAM-Representative | Volvo | | |
| Mathias Mackh<br>ASAM-Representative | ETAS GmbH | | |
| Ralf Ocker<br>ASAM-Representative | Robert Bosch GmbH | | |
| Oliver Rumpf<br>MSR-Representative | BMW AG | | |
| Markus Steffelbauer<br>ASAM-Representative | Softing | | |
| Hans Jürgen Sonnleitner<br>MSR-Representative | Siemens AT | | |

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

| Name<br>Roles | Departement | Address | Contact |
|---|---|---|---|
| Falko Stuhler<br>ASAM-Representative | Siemens AT | | |
| Ulrich Vogel<br>MSR-MEDOC-Chair | BMW AG | | |
| Bernhard Weichel<br>Robert Bosch GmbH, MSR-Representative, technial lead | Robert Bosch GmbH | | |
| Kai Werther<br>ASAM-Representative | ETAS GmbH | | |

*Version Information*

| Document Part | Editor | | | |
|---|---|---|---|---|
| | **Company** | **Version** | **State** | **Remarks** |
| 2.1<br>WD<br>8.4.2001<br>For details refer to nr. 1, Page 82 | Bernhard Weichel | | | |

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# 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* |
| *ASAM-MCD-2MC V1.x* | The considered languages resp. DTDs are marked as technical term **[type]**=*product*. |
| *ASAM* | The committees are noted as **[type]**=*organization* |
| *A2L* | Keywords from *ASAM-MCD-2MC V1.x* are marked as technical term **[type]**=*code*. |
| *ECU* | Objects in general are marked as technical terms **[type]**=*other*. This might be automotive equipments general objects such as variables etc. |

Note that an index of the technical terms is given at the end of the document.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# 1 Introduction and positioning of SGML

Basically, *SGML* is an vendor and system neutral data format. This is widely used in the industry, especially in internet technologies (HTML, XML) but also in the aerospace, pharmaceutical and automotive domain. Various consortia and standardization organizations are defining appropriate applications of *SGML*.

## 1.1 Introduction in SGML

*SGML* is an ISO standard (ISO 8879) defining a method to serialize information models. The roots of SGML are in the documentation industry where it was developed to separate the contents of a document from the particular rendition of the document.

In order to specify the contents, it is necessary to label the meaning of the information elements. This labelling then allows to assign a rendition style to particular elements.

The approach can be generalized to arbitrary data models. This allows to use the methods not only for documentation but for general data exchange. In the past two years, *SGML* was widely adopted in the computer industry, especially in form of *XML* which is a simplified application profile of *SGML*.

The general approach of *SGML* is to surround the particular information with tags specifying the meaning of of the information:

```
<address>
  <name>Bernhard Weichel</name>
  <zip>70442</zip>
  <city>Stuttgart</city>
</address>
```

In the example above, the tags always start with the character ”<”. The pure information is ”Bernhard Weichel 70442 Stuttgart”. The meaning of the information could be determined by either a specific syntax or by introducing tags as it is done by *SGML*.

As one can see, the tags always appear in nested pairs. Therefore these tags finally represent a serialized information hierarchy. In other words, the tags represent a serialized tree. The corresponding information tree to the example above is:

```
address
  |
  +- name
  |     [Bernhard Weichel]
  +- zip
  |     [70442]
  +- city
        [Stuttgart]
```

In *SGML* it is possible to predefine the possible tree structures and check if actual files match to the defined trees. Strictly speaking, it is possible to define a grammar and check if the files match to this grammar. The lexical analysis is very easy and standardized, since elements of the grammar always start with ”<”[1]. These predefined structures are called ”document type definition” (*DTD*).

[1] Within SGML it is possible to redefine also the lexical properties of the file. For further information refer to specific SGML resources.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

Files matching such a DTD are called SGML instance. Note that an SGML instance can be distributed across multiple storage media (files, memories)

Due to the standardization of *SGML*, there are tools available which support *SGML* by nature. Such kind of tools are *SGML-parser*s, *SGML-editor*s, *SGML-Converter*s. These tools can be used as a basis for domain specific applications.

## 1.1.1 The basic information elements in SGML

As mentioned above, an *SGML* file basically consists of information and the tags indicating the meaning of the information. The information about the meaning is called markup. Therefore, an SGML file basically consists of information and markup.

In general *SGML*, provides three basic concepts for markup:

Elements: Elements are indicated by a combination of starttag and endtag. Elements may consist of data as well as other elements. This nesting of elements is used to represent the information tree.

The possible nesting of elements and text is defined in the document type definition *DTD*. Hereby it is possible to define the sequence of elements as well as how often the elements may or must occur (for further details see Topic 1.1.2 The Document type definition p. 11).

Attributes: Attributes can be attached to elements and can provide more specific information such as identifiers, modifiers. Attributes are written in the starttag as a pair of name and value as illustrated in the following example:

```
<address id="wl1234" state="checked">
...
</address>
```

Attributes cannot be nested. The possible values of attributes can be specified in greater detail (value lists, basic data types) than the contents of elements.

entities: An entity is a part of an SGML instance representing specific chunks of information kept in separate memory units (files) or in specific data formats. As mentioned above, an SGML instance can be distributed across multiple storage media. This is specified using entities.

Entities are also used to specify characters which are not allowed in the text portion of an SGML file (for example ”<” is represented as ”<”).

Entities furthermore can be used to define repeated contents only once and use it multiple times. The following example illustrate this:

```
<address id="wl1234" state="checked">
  <name>Bernhard Weichel</name>
 &stuttgart;
</address>
```

In this case, the entity ”&stuttgart;” would contain the elements ”zip” and ”city”.

In conclusion, an *SGML* file consists of text and markup, which appears as elements (starting with ”<”) and entities (starting with ”&”). Therefore, the characters ”<” and ”&” (and the string ”]]>”) are reserved and must be quoted using entities.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

### 1.1.2 The Document type definition

The document type definition determines the possible information structure in a file. The following principles apply[2]:

- All elements of the same name have the same structure. In particular this establishes a context free grammar.
- Elements may be grouped. Such groups are called model groups. An example for a model group is ”*(a, (b | c), d)*”. An instance of such a modelgroup consists of ”a”, followed either by ”b” or ”c”, followed by ”d”.
- The content of an element or a model group may be a

  sequence The elements or the modelgroups may occur only in the specified order.

  alternative The elements or the model groups may occur as exclusive alternative.

- Elements in the structure can be

| | |
|---|---|
| required | The element or the model group must be there |
| optional | The element of the model group may be left out |
| one or more | The element or the model group must occur at least once but may occur as often as needed. |
| zero or more | The element or the model group may be left out or may occur as often as needed. |

#### 1.1.2.1 Graphical conventions used in DTD diagrams

Since a *DTD* represents a grammar, it can be visualized graphically as a tree. The visualization in this document uses the notation given in Figure 1 Graphical notation of DTDs p. 13.

[2] SGML provides much more means and concepts. But these are not relevant for the understanding of this document.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

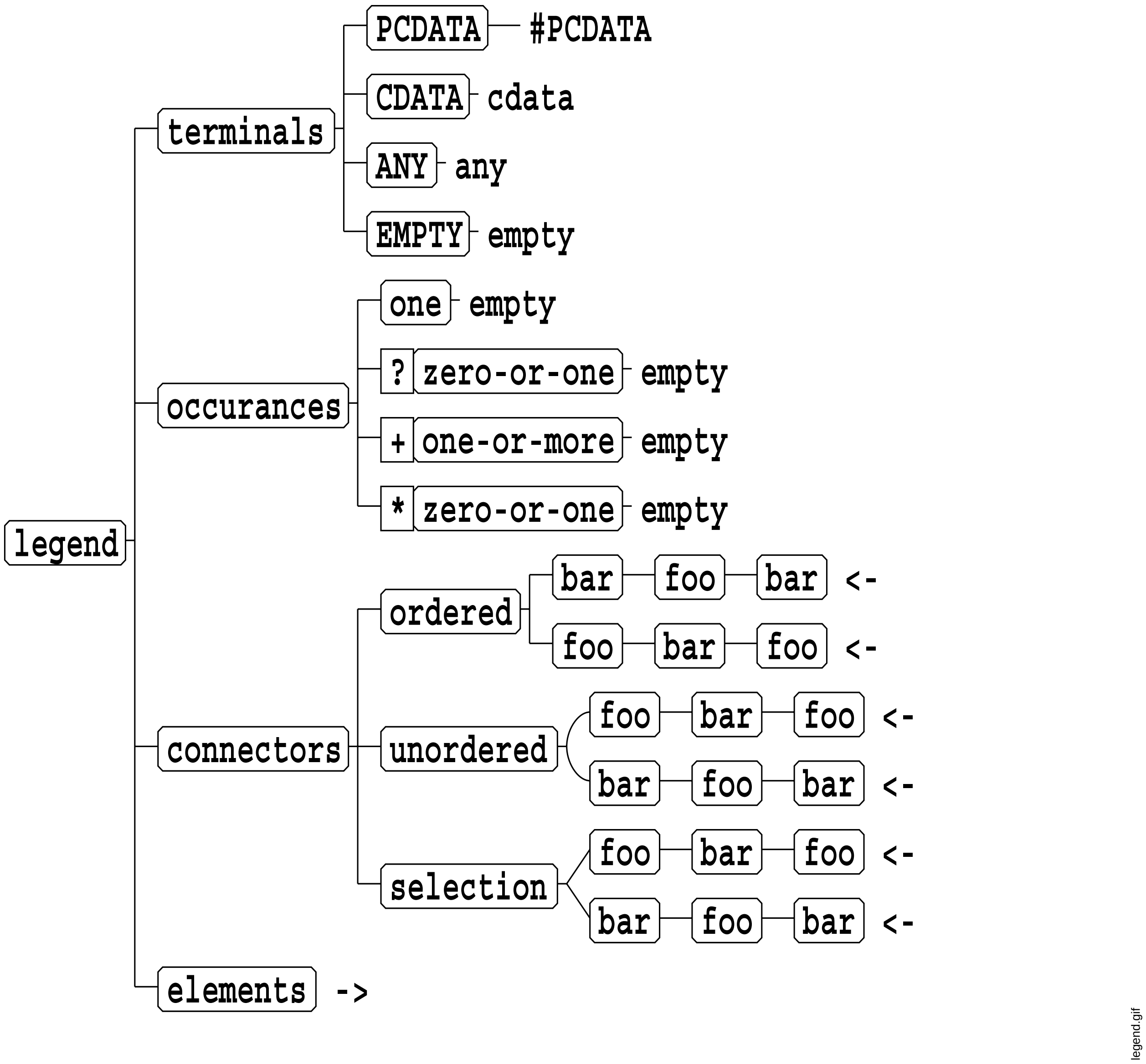



**Figure 1: Graphical notation of DTDs**

The meaning of the symbols is:

PCDATA — The element content is **Processable Character Data (PCDATA)**. This is data that consists of zero or more characters of both text and markup. 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 — The element content is **Replaceable Character Data (RCDATA)** . this is data that consists of zero or more characters, in which references

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

to substitutions are not recognized (i.e. RCDATA may contain text and entity references (starting with ”&”) , but no sub-elements)[3].

| | |
|---|---|
| CDATA | **Character Data (CDATA)** consists of zero or more text characters, where no markup of any kind is recognized. |
| ANY | a terminal type indicating that the object may contain text or any element defined in the model[4]. |
| 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 model group occurs exactly once. |
| ZERO-OR-ONE | indicates that the element or the model group is optional. |
| ONE-OR-MORE | indicates that the element or the model group occurs multiple times but at least once. |
| ZERO-OR-MORE | indicates that the element or the model 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[5]. |
| 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. |

## 1.2 Positioning SGML in a document architecture

While dealing with complex engineering processes, it is often unclear what a document really is. The three layer architecture in Figure 2 A flexible document architecture p. 15 can help in this respect since it tries to classify documents and files into specific layers.

[3] RCDATA is not used in the DTDs described here.

[4] ANY is not used in the DTDs described here.

[5] UNODERED is not used in the DTDs described here, since it is not conforming with XML.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

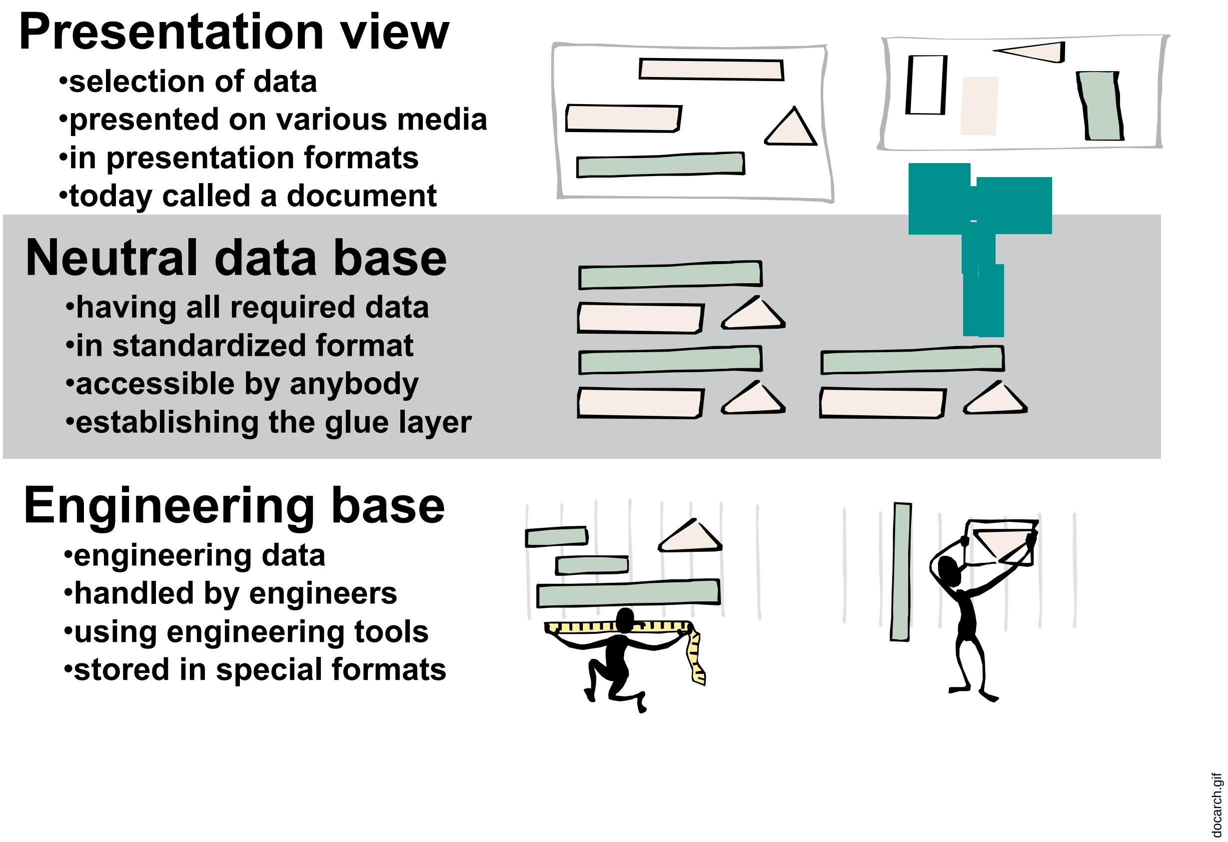



**Figure 2: A flexible document architecture**

Presentation View: This is the daily experience with a document. It is a piece of paper with a certain amount of information which is presented using various fonts, layouts and graphical elements. Progressive people use this document in electronic form.

The main focus of this layer is to present the right set of information in an appropriate layout. Therefore it is the primary domain of typewriters, word processors and electronic distribution media. It is even appropriate to position *HTML* in this layer.

Engineering Base: The engineering base is the set of engineering data in the specific data formats mainly determined by the tools used in the process. These data files are usually stored in an *Engineering data management system* which has no access to the content of the administrated data.

Nowadays engineering tools produce document views using report generators or interfaces to common word processors often proudly marketed as ”documentation interface”. In a broader sense, the in-

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

formation in the engineering base itself and not only the exported presentations must be treated as documents.

Obviously there is a gap between the presentation view and the engineering base.

Neutral Data Base

The existing gap between the two layers mentioned above can be closed by introducing a middle layer, the document base. This layer must be built upon a standard data format which can be used to generate all the required outputs (document views) in the presentation layer while still having the power to keep the semantics of the engineering base at least to some extent.

In particular the neutral data base can be used for arbitrary data exchanged if it is setup in a robust, long-term stable and powerful data format.

Within this architecture it is typical for SGML to represent the Neutral Data Base.

- Presentation formats tend to be very specific such as PDF, JPG etc. This is true even if SGML is used (in case of HTML).
- Engineering tools tend to use proprietary formats for both efficiency and data model reasons.
- The Neutral Data format must be based on standards. Efficiency is not the primary objective. Independence of the data model is important. This makes the Neutral Data Base a typical domain for SGML applications.

*ASAM-MCD-2MC V1.x* files are an approach for such a neutral data base since it is a neutral data format supported by multiple tool vendors.

The *MSR*-consortium, an initiative in the German automotive industry also works on the definition of such a neutral data base. The activities started in the documentation domain but were extended towards detailed data models.

*ASAM-MCD* and *MSR* commonly defined a data model for the domains covered by both activities. The data format is based on the ISO standard *SGML*.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# 2 MSR and ASAM-MCD

## 2.1 Introduction in MSR

*MSR* is a consortium within the German automotive industry which supports the joint development of car manufacturers and their electronic subsystem suppliers by enabling process synchronization and proper management of information exchange.

MSR establishes working groups dedicated to various fields:

*MEDOC* This working group develops standards, methods and tools for information exchange in the engineering Process. MEDOC offers unified application profiles for both data and document exchange based on the SGML/XML technology.

*MEGMA* This working group works develops methods for the exchange of simulation models in the engineering process.

*MEPRO* This working group investigates synchronization of engineering processes across multiple companies. The objective is to provide methods to define reliable synchronization points and data sets at the very beginning of a collaboration.

### 2.1.1 MSR-MEDOC - The neutral data base

This working group develops standards, methods and tools for information exchange in the engineering Process. MEDOC offers unified application profiles for both data and document exchange based on the SGML/XML technology.

This is done by defining an application profile and a set of DTDs which establishes a neutral data base (see Topic 1.2 Positioning SGML in a document architecture p. 14):

Application profile The data models defined for MEDOC are implemented as SGML/XML DTDs. All these MSR DTDs while implementing specific data models for their domain have the same basic principles:

- Same link model for all DTDs (supporting ID/IDREF, HyTime, and MSR semantic addressing simultaneously). This model allows instances of the various MSR DTDs to be linked together and used as an entire database. The link classes are unique across the entire MSR DTD system.
- basic models (generic text sections, parameters, architectures, etc.)
- configuration capabilities
- subclassing methods using < ...class> elements
- administrative data applicable to implement version control even for subtrees in one instance. This is useful if an instance is built of fragments delivered by different project partners
- Same generic approaches for constructing the DTDs (e.g. naming conventions, architectures)

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

**Reports**

When MSR started to use SGML/XML for documenting the MSR activities itself, it became clear that a DTD for arbitrary documents is required. So *MSRREP.DTD* was developed and can be used to write reports and specifications not yet covered by the other DTDs (e.g. test reports etc.)[6]. This DTD provides

- a generic chapter structure
- a detailed structure for performing change management

**Software**

For specifying and documenting software for electronic control units the *MSRSW.DTD* is used. This DTD is successfully used in projects across companies as well as across different business units in one company while each partner uses different engineering tools and SGML/XML tools.

This DTD was the SGML-basis for the joint work between *ASAM* and *MSR*.

**FMEA**

In the area of Failure Mode and Effect analysis, an experimental DTD (*MSRFMEA.DTD*) was developed. Its design is totally database oriented. The data model behind it was influenced by the data model of an existing tool for FMEA. This data model was generalized to eliminate the tool specifics.

**Diagnosis**

The latest activity is the documentation for diagnosis (on board as well as off board). The work started in spring 98. This domain touches all the other DTDs. So we expect all the information being available in instances of MSRSYS, MSRNET, MSRSW, MSRFMEA. Nevertheless an integrated document set for diagnosis is required. For configuration of off board test systems, an SGML DTD is defined by *ASAM-MCD-2D* . This DTD is compatible with the strategies of MSR.

**Networks**

For specification and documentation of on board networks, the *MSRNET.DTD* is used. It allows to specify the information transported on the network, packed these signals into messages the network topology etc. This DTD is implemented as import/export facility in the CAN tools from CAN Vector Informatik.

**System**

MSR started in the domain of entire systems. The result is the *MSRSYS.DTD* which is used to describe and to specify entire control systems with all its mechanical and electrical components. This DTD provides detailed structures for:

- project data
- parts and system decomposition
- architectures with signal, interfaces, ports and connection specification
- connections
- electrical characteristics

[6] The document you are reading is also written using MSRREP.DTD.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

- mechanical characteristics
- optical and acoustical characteristics
- environmental characteristics

## 2.2 ASAM-introduction

*ASAP* is an initiative to standardize interfaces in the the domain of measurement and calibration of electronic control systems in the automotive industry. Since 1999, the activities are driven by a newly founded association called *ASAM e.V* (**A**ssociation for **S**tandardisation of **A**utomation- and **M**easuring Systems).

The individual application systems (AS) of the measurement, application and diagnostic system (MCD) are linked to the automation via interface *ASAM-MCD MC3*, they obtain information about the control unit's internal elements, its interfaces and communication methods from the *ASAM-MCD MC2* description file, and are in turn linked to the control units (ECU) and the control unit dependent measurement technology (ADC) via the *ASAM-MCD 1b* interface via ROM emulators, CAN or ABUS or the diagnostic bus (D bus). This structure allows current monolithic applications to be divided into compatible subsystems.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

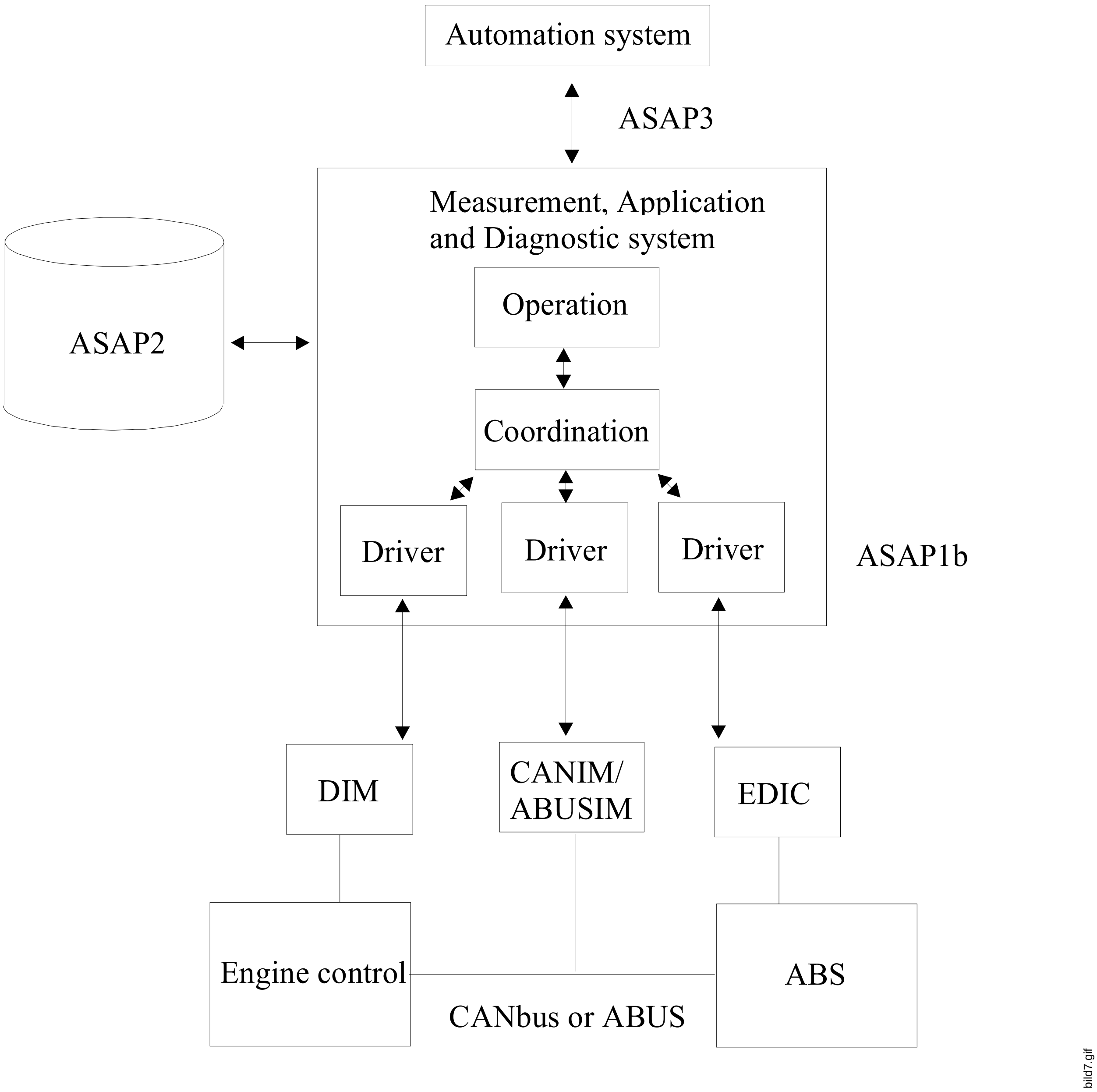



**Figure 3: Overall system and interfaces**

Via the *ASAM-MCD 1b* interface the standard connection of the control units and of the control unit dependent measurement technology is implemented independently of the chosen communication path or the relevant supplier of the control unit. To obtain this functionality, the control unit or the measurement system is linked to the measurement, application and diagnostic system via the transport path, the interface hardware and a driver in accordance with the ASAM-MCD specifications. This subsystem below interface

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

1b is identified by the ASAM-MCD device (Figure 4 ASAM-MCD device (principle structure) p. 21):

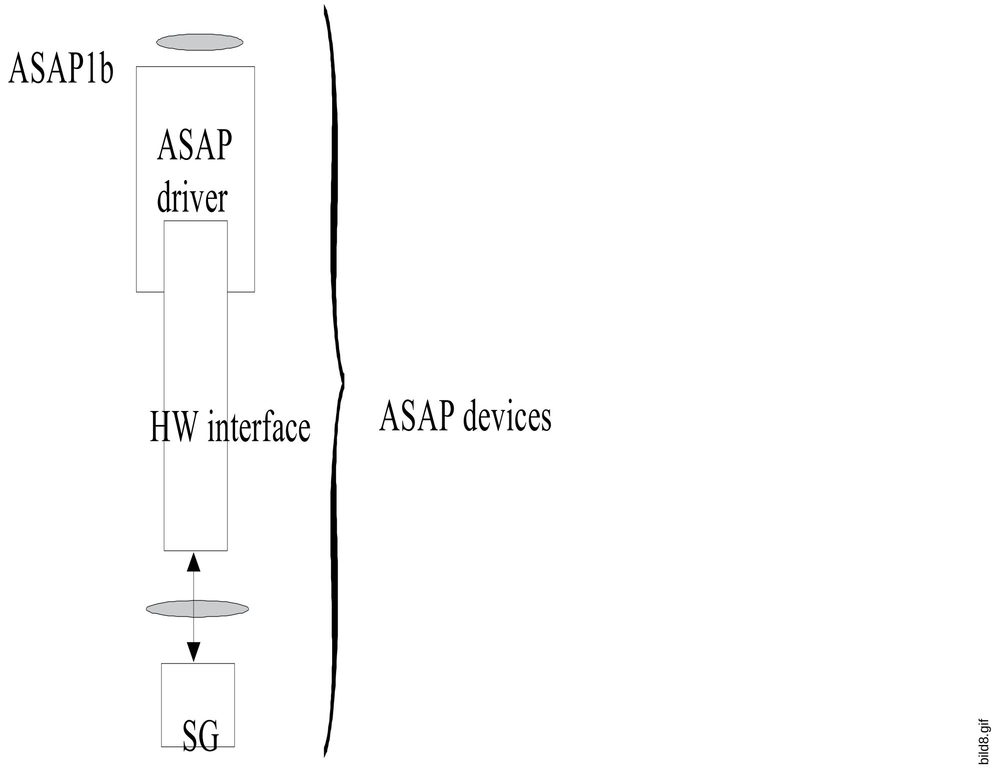



**Figure 4: ASAM-MCD device (principle structure)**

An ASAM-MCD device thus defined may be composed of different elements depending on the selected connection (Figure 5 Various versions of the ASAM-MCD device. p. 22).

The *ASAM-MCD 1b* interface is a functional interface which was initially defined independently of the MCD operating system. It offers a range of services controlling the exchange of parameters and data via the *ASAM-MCD 1b*.

This allows similar subsystems such as ROM emulators of the firms AB and XY to be accessed in the same way. This 'similarity' is related to the use of the services which imply above all commonly agreed communication procedures. Special features and different communication methods within the ASAM-MCD device are embedded in this device, the setting of parameters occurs by allocating equally special binary objects and parameters from the ECU description file. Interpretation of the objects is only possible by the ASAM-MCD device. Furthermore, uniform communication with the control unit is possible independently of the selected connection. Different features of the various control unit connections can be balanced within the ASAM-MCD device, while the functional communication, e.g. the setting of parameters, is also standardized.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

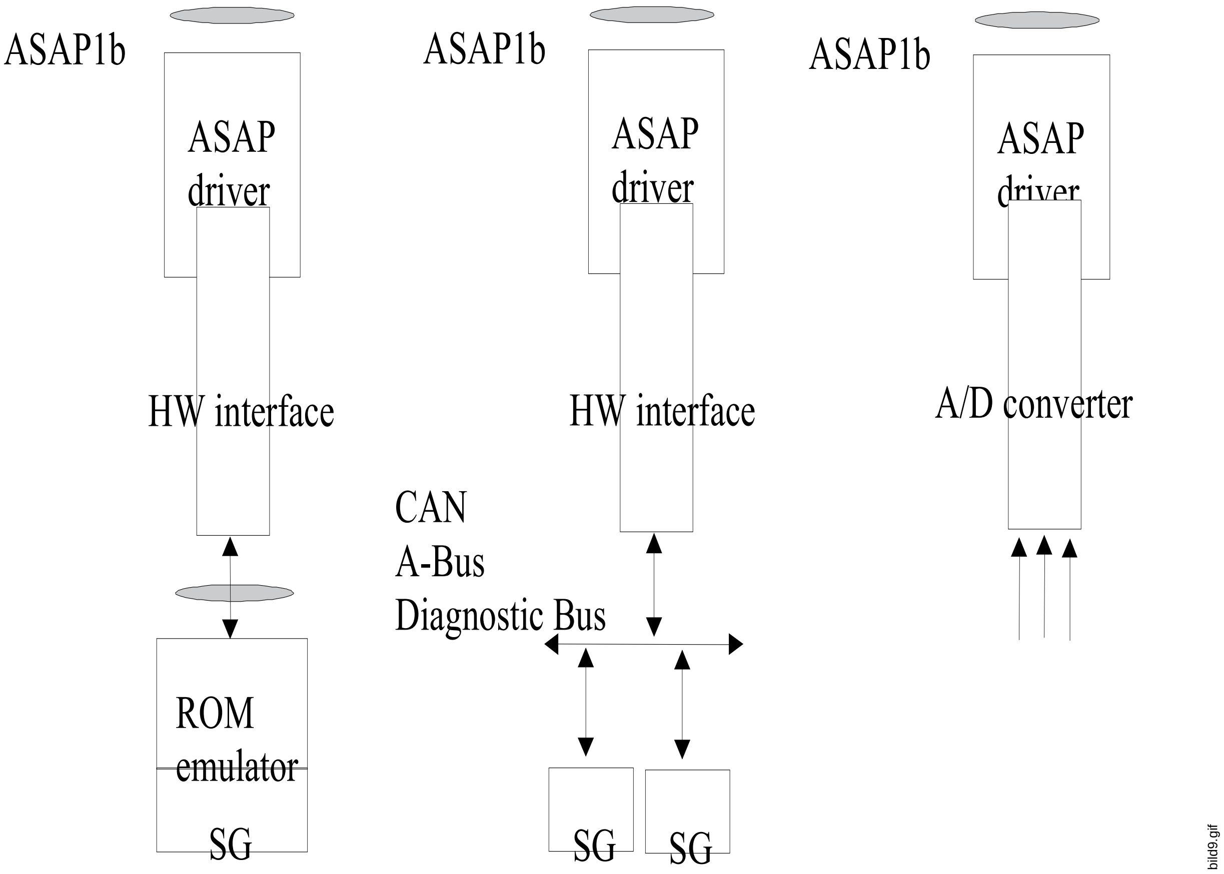



**Figure 5: Various versions of the ASAM-MCD device.**

The ASAM-MCD description file (*ASAM-MCD MC2*) is used to describe the ECU internal data, constitutes the reference for an individual control unit and its link to the *ASAM-MCD 1b*.

Interface *ASAM-MCD MC3* links up the measurement, application and diagnostic system (MCD) to an automation system (*AuSy*). From the standpoint of the *AuSy* it can therefore be considered as an intelligent device as e.g. an indexing device or a fuel scale. It incorporates both the various methods for access to control units as well as the individual structures in the control units and offers higher-level functions.

The *AuSy* does not require any special knowledge about the methodology and the parameter setting of the interfaces to the control units for its automation sequences. This service is provided by the MCD. The connection is established via the *ASAM-MCD MC3* interface.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# 3 Basic Data model regarding *ASAM-MCD-2MC* usecase

The MSR software dtd is designed for describing the complete development cycle for embedded control software. The application relevant issues have been introduced like record layouts, MC-interfaces (blobs). Other issues have been refined like the definition of base types and the semantic definition of physical units.

Although *MSRSW.DTD* is designed for a rich set of usecaes, the *ASAM-MCD-2MC* usecase is displayed here in particular. The basic structure is given in Figure 6 Basic structure of ASAM-MCD-2MC usecase p. 24.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

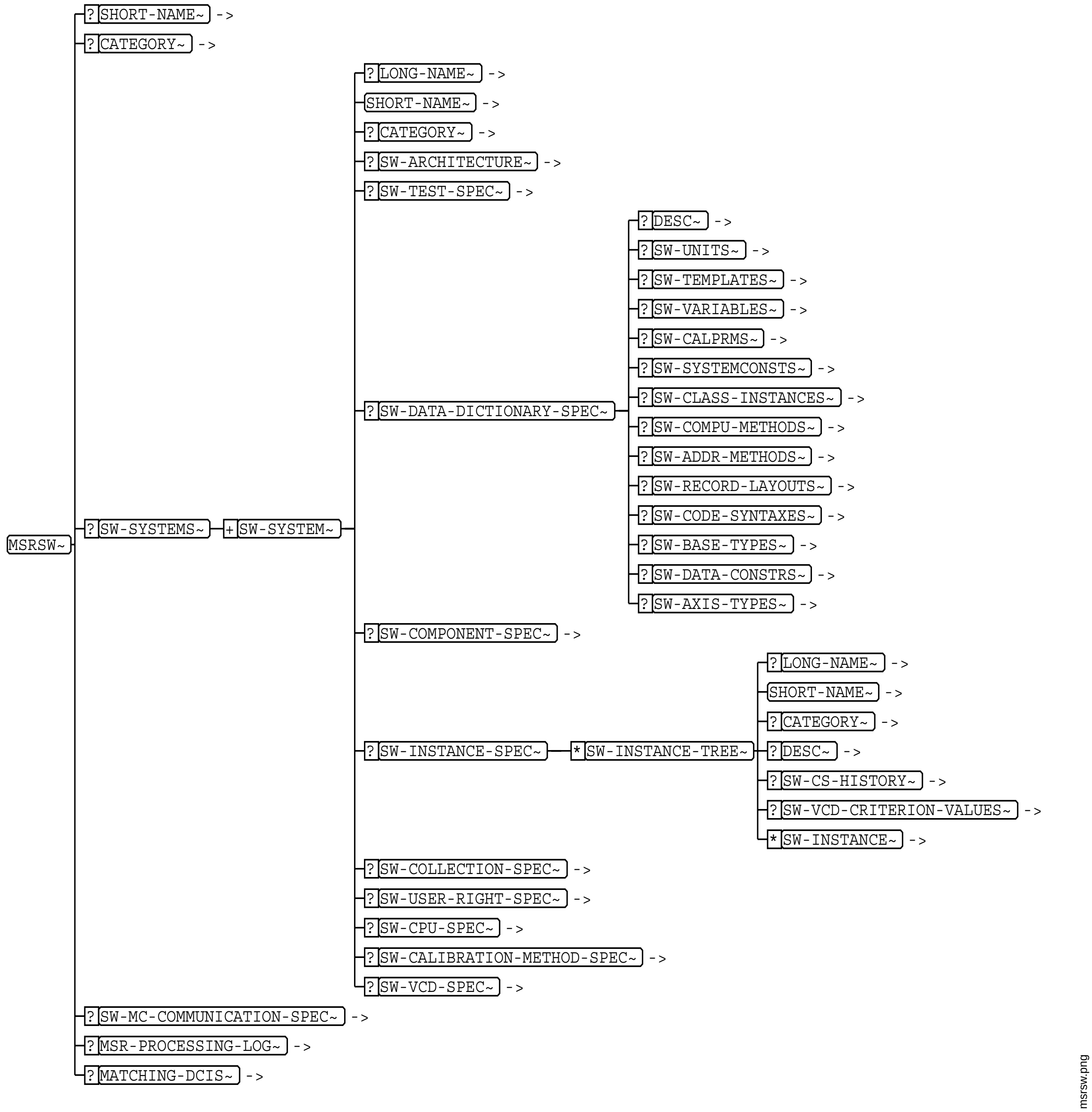



**Figure 6: Basic structure of ASAM-MCD-2MC usecase**

The basic description elements are the same as in *ASAM-MCD-2MC V1.x* like measurements, characteristics and computation methods, however the respective notion from MSR software (*MSRSW.DTD*) has been taken over.

The usage of data dictionaries allows a structuring of the description with respect to functionality or substructuring. In addition there are further extensions that although coming

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

from other process phases (like software design and coding) can be used in the calibration process too.

One important extension is that the properties of an element can be described in different places, which allows the reuse of such information (e.g. if a structure appears twice, a good portion of the property is the same for both structured can be shared). Furthermore the fact that both are derived from the same source can explicitely be stated. This being crucial for early phases in the process where a bidirectional exchange is necessary, can have a benefit in the calibration business too. For more about this road ahead see Topic 7 The Road ahead p. 62.

## 3.1 The data dictionary

The data dictionary **<sw-data-dictionary>** basically contains the definition of computation methods (**<sw-compu-method>**), variables (**<sw-variable>**) and parameters (**<sw-param>**). In an MSR file there can be global data dictionary and function[7] (**<sw-feature>**) specific local data dictionaries. This allows to (semantically) describe the structure of the software and to have function local name spaces. This is a major extension to *ASAM-MCD-2MC V1.x*, which has only one global data dictionary with only one global name space. Note that more detailled data model in UML-notation is provided in Topic 7.3 More details on data models p. 66.

[7] For sake of clarity, the workgroup decided to talk fo features instead of functions which have been misunderstood as mathematical functions

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

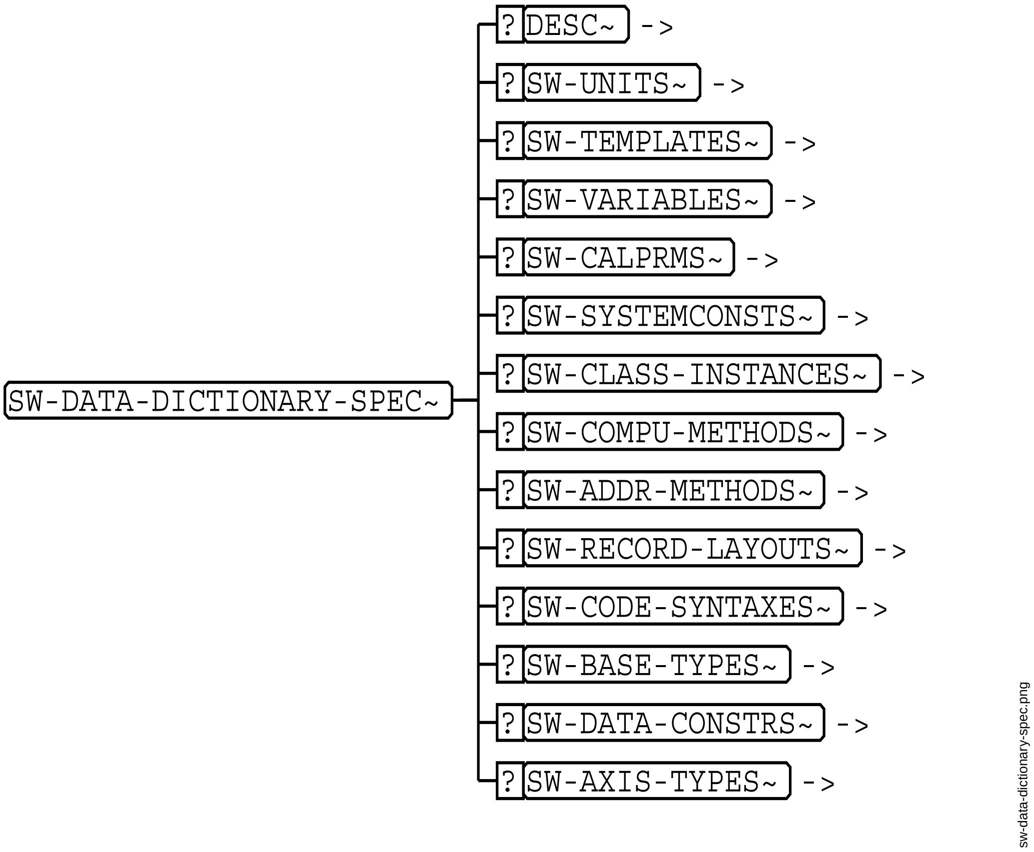



**Figure 7: sw-data-dictionary**

In the "default" conversion of *ASAM-MCD-2MC V1.x* to *ASAM-MCD-2MC2V2.x* the possiblity of local data dictionaries will not be used, but everything will be placed into the one global data dictionary.

Another major enhancement in the concepts of *ASAM-MCD-2MC V2.0* vs. *ASAM-MCD-2MC V1.x* and *MSRSW V1.1.0*, and hopefully also a great improvement is the seperate treatment of a kind of "type definition"[8] and the instantiation of actual variables (measurements) and parameters (characteristics). In *ASAM-MCD-2MC V1.x* no such difference can be seen, since only the instances (with all instance specific information like address or interface data) appear here. Other information, that are held at each measurement or parameter, are derived from some kind of type and are repeated for each measurement and characteristic of that type (e.g. all velocities have the same formula, range, etc.) More over, all instances are global here. From the perspective of calibration and measurement this approach seems to be well suited.

The *ASAM-MCD-2MC V2.0* belongs to the "family" of MSRSW DTD instances (like e.g. also *PACO* etc.), where each member serves a specific purpose in the development process. In the function and software development phase also the structure must be described in such

[8] sometimes called data structure definition

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

files since this information is vital here. To satisfy these needs the data specific information is distributed in the data dictionary and in the instance tree.

Basically the instance **<sw-instance>**tree refers to what can be seen the instance list in *ASAM-MCD-2MC V1.3*.

### 3.1.1 Base types

Base types can be formally specified in *ASAM-MCD-2MC V2.x*. The base types used in the measurements (variables) and characteristics (parameters) reference this specified (and freely named) base types. This was not possible in *ASAM-MCD-2MC V1.x* and strengthens the semantic thoroughness of the software description.

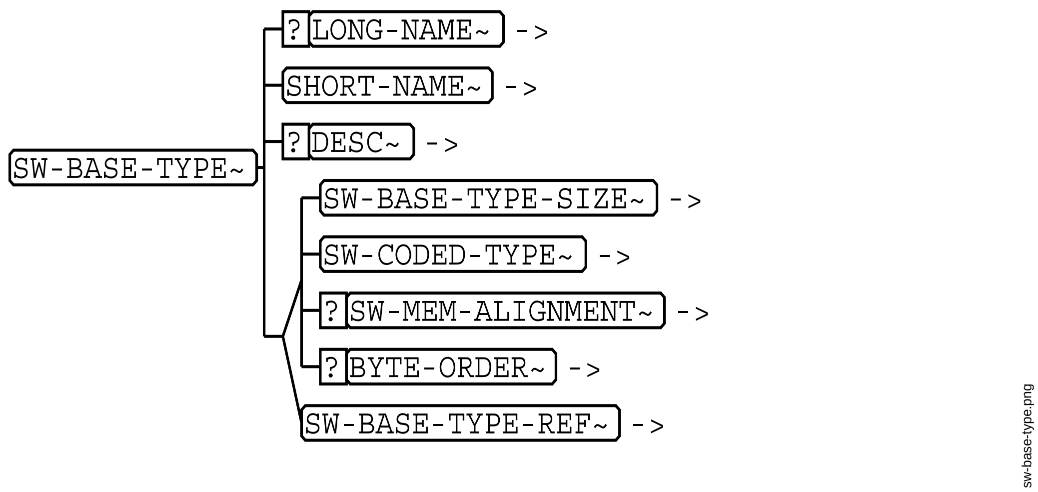



**Figure 8: sw-base-type**

On the other hand, this eliminates the predefined data types of *ASAM-MCD-2MC V1.x*. The following example is taken from the sample instance below (Topic 5.3.1.7 Base types for the CPU p. 47).

```
<sw-base-type id="UBYTE">
   <long-name>Unsigned byte</long-name>
   <short-name>ubyte</short-name>
   <sw-base-type-size>8</sw-base-type-size>
   <sw-coded-type>unsigned</sw-coded-type>
   <sw-mem-alignment>8</sw-mem-alignment>
   <byte-order type="MOST-SIGNIFICANT-BYTE-LAST"></byte-order>
</sw-base-type>
```

### 3.1.2 Physical units

Physical units are crucial for the correct interpretation of a numerical value. One of the most popular errors due to the wrong interpretation of physical metrics was the simultaneous use of the US-american metric system (feet) and the standardised metric system (meter) in the space industry (Pathfinder II) which caused a loss of approx. one billion dollars.

In *ASAM-MCD-2MC V1.x* the physical units are described simply by a string without physical meaning behind this.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

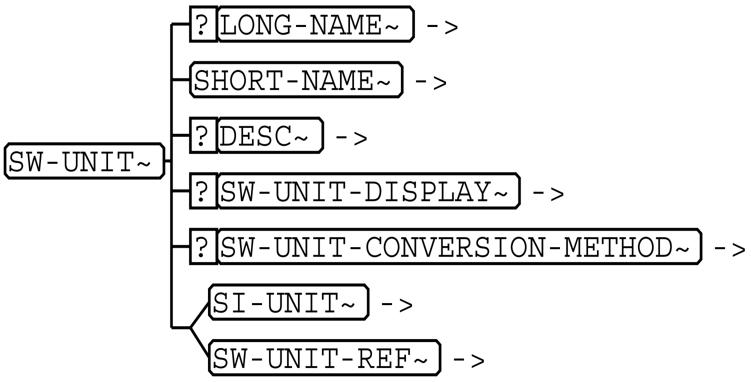



**Figure 9: sw-unit**

The principle of describing a "real" physical unit is to refer to the SI (standardisation international)-units. Here the seven base dimensions are given (length, time, mass, temperature, current, luminous intensity, amount of substance) with their respective base unit (meter, second, kilogramm, etc.). Each physical dimension can be expressed as a product/quotient of these dimensions with its corresponding SI-unit.

Each other unit is in a linear relationship to the corresponding SI-unit or other units of the same physical dimension, e.g. feet = 0.3048 * meters, Celsius = Kelvin - 273,15. An example of how this is described is given below (more examples see Topic 5.3.1.1 Measurement units p. 38):

```
<sw-unit id="KELVIN">
  <long-name>Kelvin</long-name>
  <short-name>kelvin</short-name>
  <sw-unit-display>K</sw-unit-display>
  <si-unit thermodynamic-temperature-expo="1"></si-unit>
</sw-unit>

<sw-unit id="GRAD-C">
  <long-name>Grad C</long-name>
  <short-name>Grad-C</short-name>
  <sw-unit-display>Grad C</sw-unit-display>
  <sw-unit-conversion-method>
   <sw-unit-gradient>1</sw-unit-gradient>
   <sw-unit-offset>-273.15</sw-unit-offset>
  </sw-unit-conversion-method>
  <sw-unit-ref id-ref="KELVIN">KELVIN</sw-unit-ref>
</sw-unit>
```

The **<sw-unit-conversion-method>** describes, how the acutal unit depends on the referenced unit. In particular, the actual unit is a function of the referenced unit:

```
[Grad C] = 1 * [kelvin] - 273
```

## 3.1.3 Computation methods

The data model for computation methods has not changed substantially from *ASAM-MCD-2MC V1.x* to *ASAM-MCD-2MC V2.x*. So guys, be happy, not all of the good old stuff is gone.

The major changes are:

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

- generalized rational computation formula
- reference to (semantic) units
- unification and generalization of "*VTAB*" and "*VTAB_RANGE*"

The structure of computation methods can be seen in Figure 10 sw-compu-method p. 29

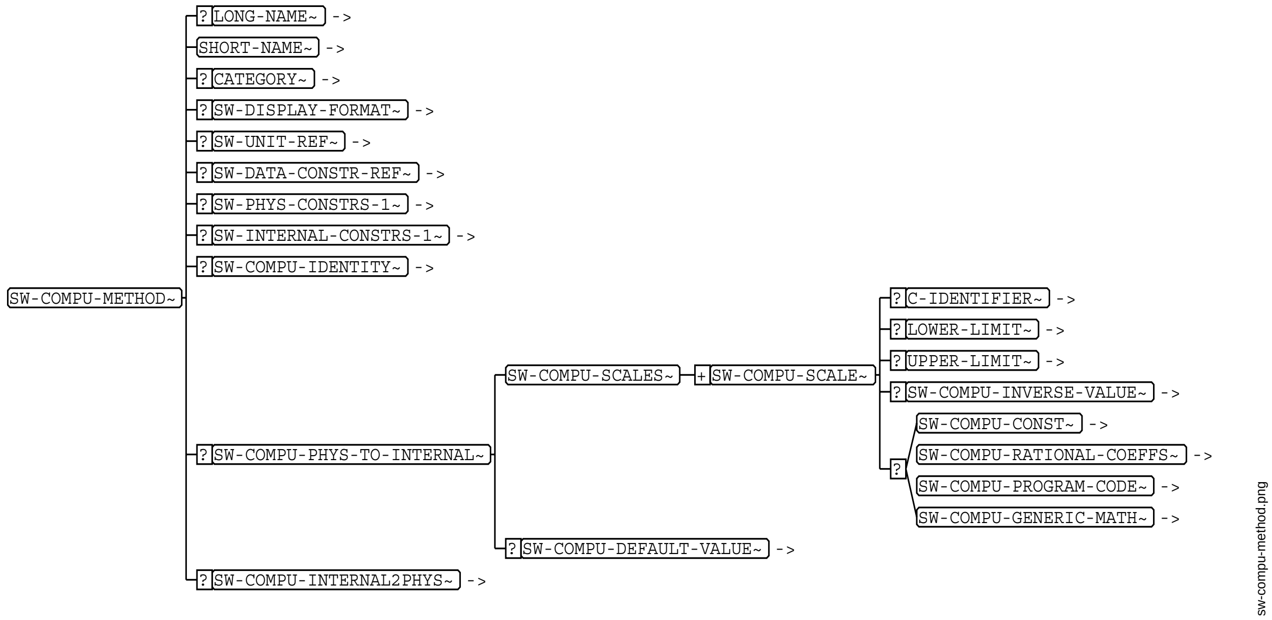


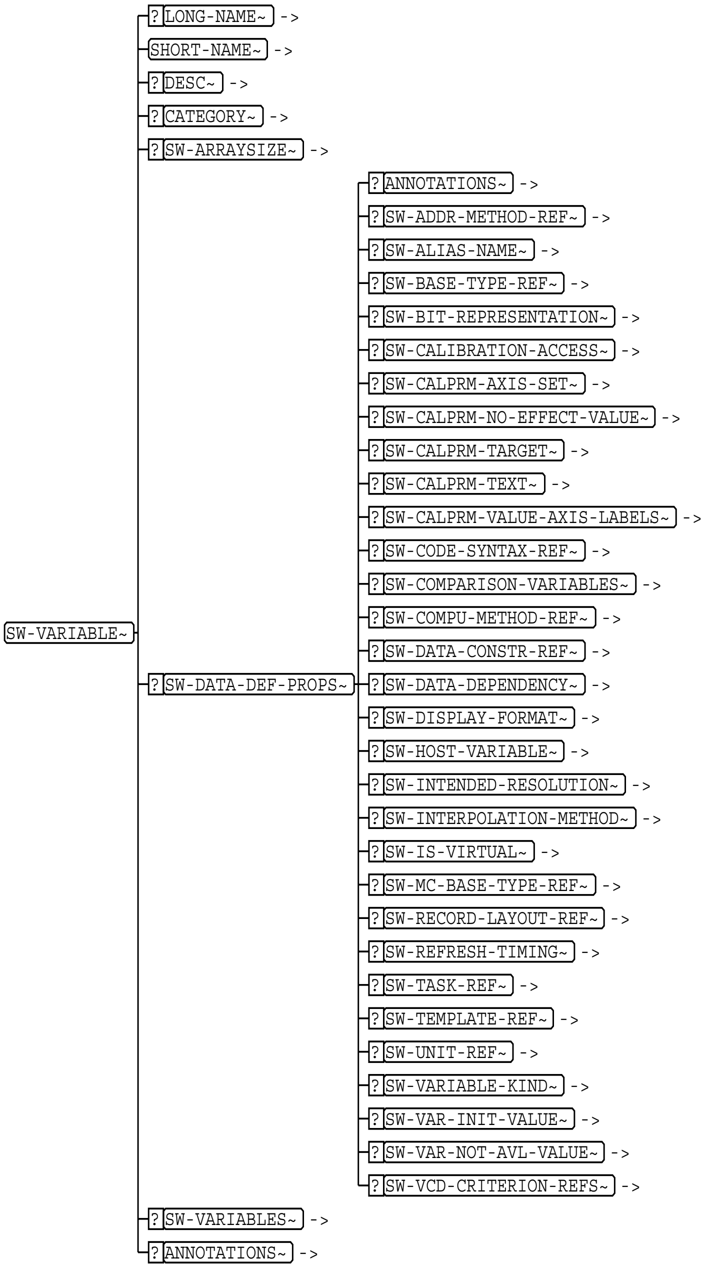

**Figure 10: sw-compu-method**

## 3.1.4 Variables and parameters

In *ASAM-MCD-2MC V1.x* there *MEASUREMENTS* and *CHARACTERISTICS*. In *ASAM-MCD-2MC V2.0* (due to its other parent *MSRSW .DTD*) the notions variable (**<sw-variable>**) and parameter **<sw-param>** are used for measurements and characteristics. Both have a very similar but slightly different description, which has already lead to some confusion (e.g. adaptable characteristic maps). Therefore the content model between both has been unified.

The property of a variable can be described at various levels (places in the *ASAM-MCD-2MCV2.0* file). For the ASAM-MCD-2MC use case the data dictionary and the instance tree are the relevant places, but it seems advisable to describe only the most instance-specific properties like address at the instance level.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.




sw-variable.png

**Figure 11: sw-variable**

## 3.1.5 Record Layouts

In *ASAM-MCD-2MC V1.x* the description of the record layout is one of the most important topics (as can be seen from the frequent changes in this area). It has shown, that the approach in *ASAM-MCD-2MC V1.x* was to restrictive. Therefore a generalized approach was taken in *ASAM-MCD-2MC V2.x*.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

Major improvements are

- arbitrary dimensions without additional keywords
- iterators allowing flexible record layout structure.

Very illustrative examples and comments can be seen in Topic 5.3.1.6 Record layouts p. 44

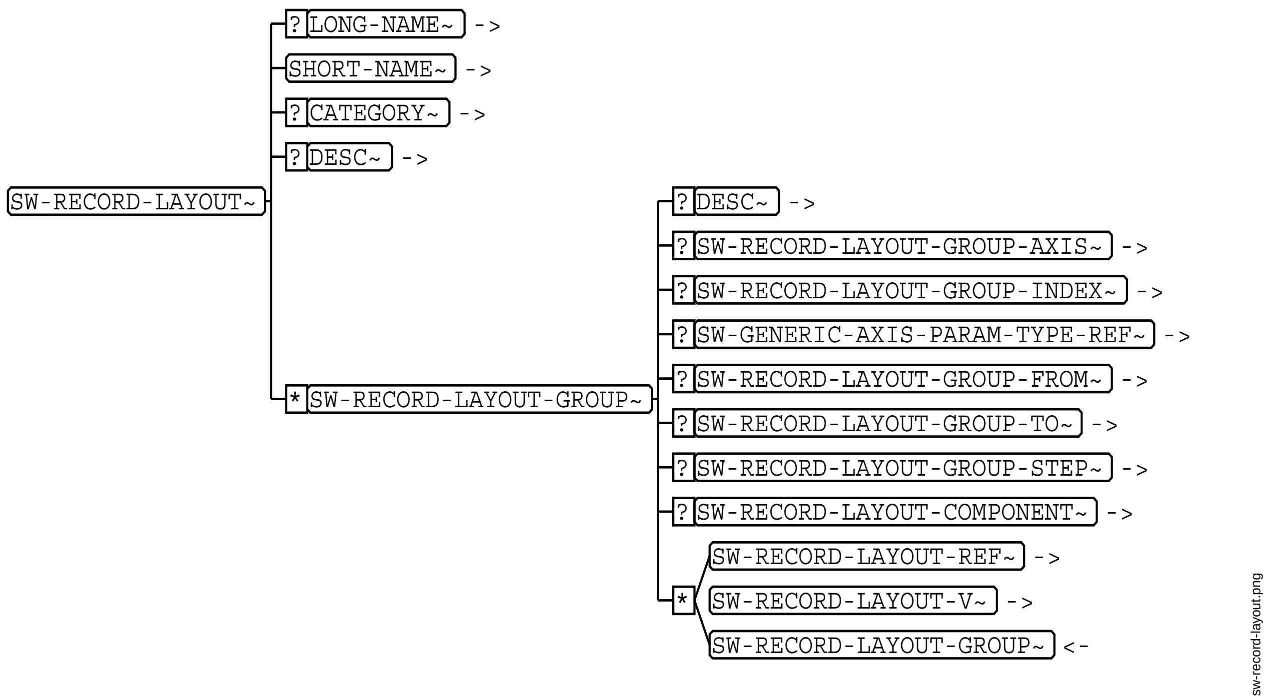



**Figure 12: sw-record-layout**

## 3.2 Collections

*ASAM-MCD-2MC V1.3* introduces new keyword *GROUP*, *SUB_GROUP*, *REF_GROUP ROOT*. *GROUP*. These keywords have been generalized as mechanism to define collections which can be used for arbitrary purposes. In principle, a collection comprises of:

- a set of rule **<sw-collection-rules>**. This is defined as wildcard, regular expression or scripts.
- the final contents of the collection (**<sw-collection-contents>**) which is the result of applying the **<sw-collection-rules>** to the database.

There may be several motivations to define groups such as

- application of access rights
- building work packages for the calibration phase
- assigning responsibilities
- general data management
- Selection criteria in an *MCD-System*.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

## 3.3 The instance-tree

The instance tree contains all ”real existing” instances, i.e. variables and parameters, that are part of the ECU software. The structure is in the form of a tree (in contrast to *ASAM-MCD-2MC V1.3* where only a flat list is possible) to reflect the natural (e.g. function resp. feature) structure.

At the instance level properties can be inherited by the ”type”, i.e. from an entry in a data dictionary. This is realized by a reference from an instance to an entry in the data dictionary. This properties of an instance are than enriched or overwritten by additional instance specific properties.

This sharing allows to reuse information like it is common in programming. In a ”default” conversion between *ASAM-MCD-2MC V1.3* and *ASAM-MCD-2MC V2.0* such structuring informaion however will not be taken into account (since *ASAM-MCD-2MC V1.3* is not able to express it).

### 3.3.1 The return of the Blobs

An important part of the *ASAM-MCD-2MC V1.x* is to describe the ECU interface to the calibration system. Since this interface itself is not really part of the description of the ECU data structures and is subject to frequent changes, it has been put into the *IFDATA* sections.

In order to guarantee compatiblity with the *ASAM-MCD-2MC V1.x* also the MC-Interface can be described in *ASAM-MCD-2MC V2.x*. Like in *ASAM-MCD-2MC V1.x* the MC-Interface description is split into two parts, one containing initialization and source information, the other describing the access to actual data.

Both parts are now clearly separated. The first is part of the **<sw-mc-communication-spec>**, the second is part of the instance tree (**<sw-instance-spec>**).

A major drawback of the interface specific information contained in the *ASAM-MCD-2MC V1.3* format is the processing of the data since it is very specific and causes problems in the software production process. Also information of the access blobs is often highly redundant to the information already contained in the *ASAM-MCD-2MC V1.x* description and can be derived from that.

In order to overcome these difficulties a generic approach has been discussed but finally could not be worked out to its full extent (called general resolver). In order to provide a solution to the current approach, elements are there to provide the blob layout in microsyntax according to *ASAM-MCD-2MC V1.3* (e.g. **<sw-mc-kp-blob-layout>**) as well as element for the blob contents also in microsyntax according to *ASAM-MCD-2MC V1.3* (e.g. **<sw-mc-kp-blob-conts>**). These elements may be filled with text only.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# 4 Sample correspondance between *ASAM-MCD-2MC V1.x* and *ASAM-MCD-2MC V2.0*

In the following figures[9] an attempt is made to illustrate how *ASAM-MCD-2MC V1.x* and *ASAM-MCD-2MC V2.0* correspond to each other. Note that the SGML data is sometimes expressed as XML especially for empty elements. This is only a syntactic issue and has no influence to the semantic.

The examples are taken from the sample files given in Topic 5 Annotated Sample file p. 37 and Topic 6 The corresponding ASAM-MCD-2MC File p. 55.

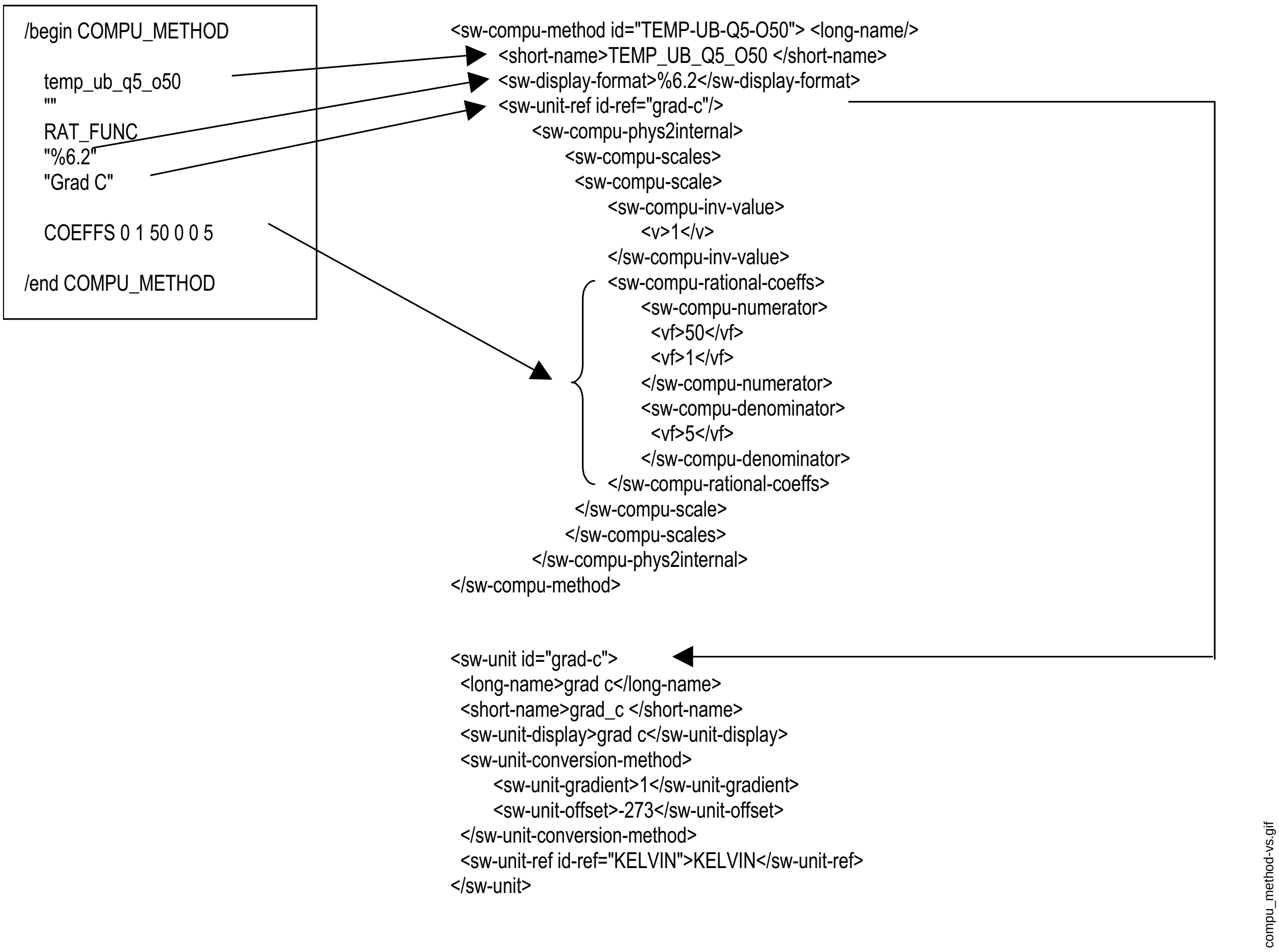



**Figure 13: Computation method**

[9] These figures still reflect the prerelease 4 of ASAM-2MCD-MC.DTD

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

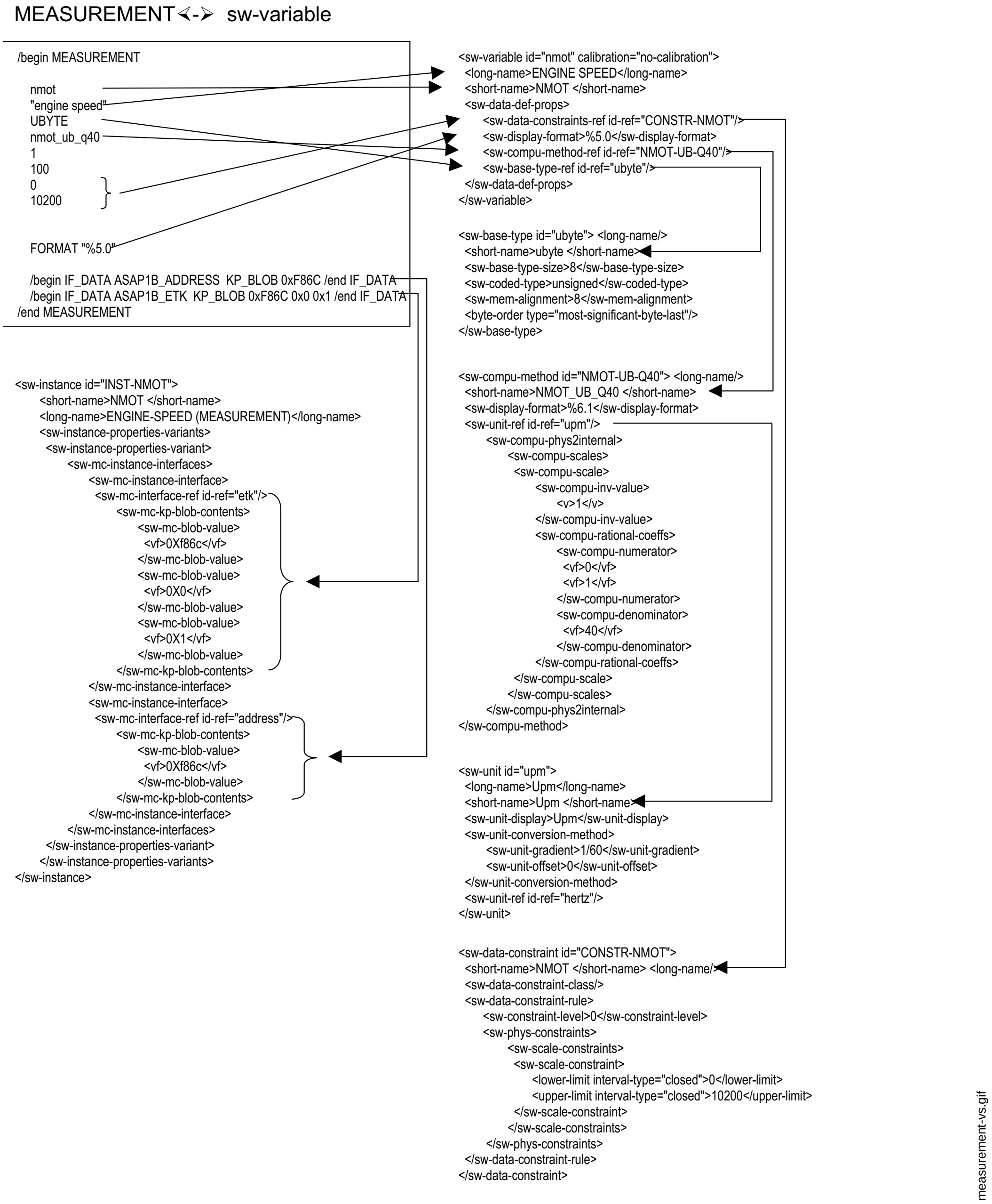



**Figure 14: Measurements**

A2ML<-> sw-mc-interface-spec

```
/begin A2ML
   taggedunion if_data {

/*****************************************************************/
/*                                                                */
/*   ASAP2-Meta-Beschreibung für ETK                              */
/*   - nur in ASAP V1.2 anwendbar -                               */
/*                                                                */
/*   Robert Bosch GmbH K3/EES4-Hünerfeld                          */
/*   Stand 0.2, 12.Aug.96                                         */
/*   Stand 0.3, 28.05.97 ASAP1B_ADDRESS                           */
/*                                                                */
/*****************************************************************/

"ASAP1B_ETK" taggedstruct
{
 (block "SOURCE" struct
  {
   struct
   {
    char [100]; /* Source-Name */
    int;    /* Minimale Periodendauer gemessen in CSE */
    long;   /* Minimale Anzahl von Perioden */
   };
   taggedstruct
   {
    "QP_BLOB" struct   /* QP_BLOB fuer ASAP1b */
    {
     int; /* Max. Groesse Displaytabelle */
     long;      /* Adresse Displaytabelle */
    };
   };
  };
 ) *;        /* SOURCE kann mehrfach auftreten */

 "TP_BLOB" struct /* TP_BLOB fuer ASAP1b */
 {
  long;   /* Trigger-Segment-Adresse */
  int;       /* Ausgabeformat der Displaytabelle 1=byte 2=word */
  int;       /* Trigger-Modus (spezielle Codierung) */
  int;       /* Typ der Displaytabelle (12, 11, 20) */
  int;       /* Byte-Reihenfolge 1 = high first, 2 = low first */
 };

 "DP_BLOB" struct /* DP_BLOB fuer ASAP1b */
 {
  long;   /* Anfangs-Adresse */
  long;   /* Laenge */
 };

 "KP_BLOB" struct /* KP_BLOB fuer ASAP1b */
 {
  long;   /* Anfangs-Adresse */
  int;       /* Adress-Ort (0=intern, 1=extern) */
  int;       /* Anzahl der Bytes */
 };

};

"ASAP1B_ADDRESS" taggedstruct
{
 "KP_BLOB" struct /* KP_BLOB fuer ASAP1b */
 {
  long;   /* Anfangs-Adresse */
 };

};

};
 /end A2ML
```

```
<sw-mc-interface id="etk">
 <short-name>etk </short-name>
 <sw-mc-blob-layouts>
    <sw-mc-tp-blob-layout>
       <sw-mc-blob-entry id="TP-TRIG-SEG-ADR">
        <short-name>TP_TRIG_SEG_ADR </short-name>
        <sw-mc-base-type-ref id-ref="LONG"/>
       </sw-mc-blob-entry>
       <sw-mc-blob-entry id="TP-DISP-FORMAT">
        <short-name>TP_DISP_FORMAT </short-name>
        <sw-mc-base-type-ref id-ref="INT"/>
       </sw-mc-blob-entry>
       <sw-mc-blob-entry id="TP-TRIG-MODE">
        <short-name>TP_TRIG_MODE </short-name>
        <sw-mc-base-type-ref id-ref="INT"/>
       </sw-mc-blob-entry>
       <sw-mc-blob-entry id="TP-DISP-TYPE">
        <short-name>TP_DISP_TYPE </short-name>
        <sw-mc-base-type-ref id-ref="INT"/>
       </sw-mc-blob-entry>
       <sw-mc-blob-entry id="TP-BYTE-ALIGNMENT">
        <short-name>TP_BYTE_ALIGNMENT </short-name>
        <sw-mc-base-type-ref id-ref="INT"/>
       </sw-mc-blob-entry>
    </sw-mc-tp-blob-layout>
    <sw-mc-qp-blob-layout>
       <sw-mc-blob-entry id="QP-DISP-SIZE">
        <short-name>QP_DISP_SIZE </short-name>
        <sw-mc-base-type-ref id-ref="INT"/>
       </sw-mc-blob-entry>
       <sw-mc-blob-entry id="QP-DISP-ADR">
        <short-name>QP_DISP_ADR </short-name>
        <sw-mc-base-type-ref id-ref="LONG"/>
       </sw-mc-blob-entry>
    </sw-mc-qp-blob-layout>
    <sw-mc-kp-blob-layout>
       <sw-mc-blob-entry id="ETK-KP-ADR">
        <short-name>KP_ADR </short-name>
        <sw-mc-base-type-ref id-ref="LONG"/>
       </sw-mc-blob-entry>
       <sw-mc-blob-entry id="KP-ADR-LOC">
        <short-name>KP_ADR_LOC </short-name>
        <sw-mc-base-type-ref id-ref="INT"/>
       </sw-mc-blob-entry>
       <sw-mc-blob-entry id="KP-BYTE-COUNT">
        <short-name>KP_BYTE_COUNT </short-name>
        <sw-mc-base-type-ref id-ref="INT"/>
       </sw-mc-blob-entry>
    </sw-mc-kp-blob-layout>
    <sw-mc-dp-blob-layout>
       <sw-mc-blob-entry id="DP-ADR">
        <short-name>DP_ADR </short-name>
        <sw-mc-base-type-ref id-ref="LONG"/>
       </sw-mc-blob-entry>
       <sw-mc-blob-entry id="DP-LEN">
        <short-name>DP_LEN </short-name>
        <sw-mc-base-type-ref id-ref="LONG"/>
       </sw-mc-blob-entry>
    </sw-mc-dp-blob-layout>
 </sw-mc-blob-layouts>
</sw-mc-interface>

<sw-mc-interface id="address">
 <short-name>address </short-name>
 <sw-mc-blob-layouts>
    <sw-mc-kp-blob-layout>
       <sw-mc-blob-entry id="ADRESS-KP-ADR">
        <short-name>KP_ADR </short-name>
        <sw-mc-base-type-ref id-ref="LONG"/>
       </sw-mc-blob-entry>
    </sw-mc-kp-blob-layout>
 </sw-mc-blob-layouts>
</sw-mc-interface>
```

a2ml-vs.gif

**Figure 15: interface to mc-system (a2l)**

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

IF_DATA ⮜-⮞ sw-mc-interface-implementation

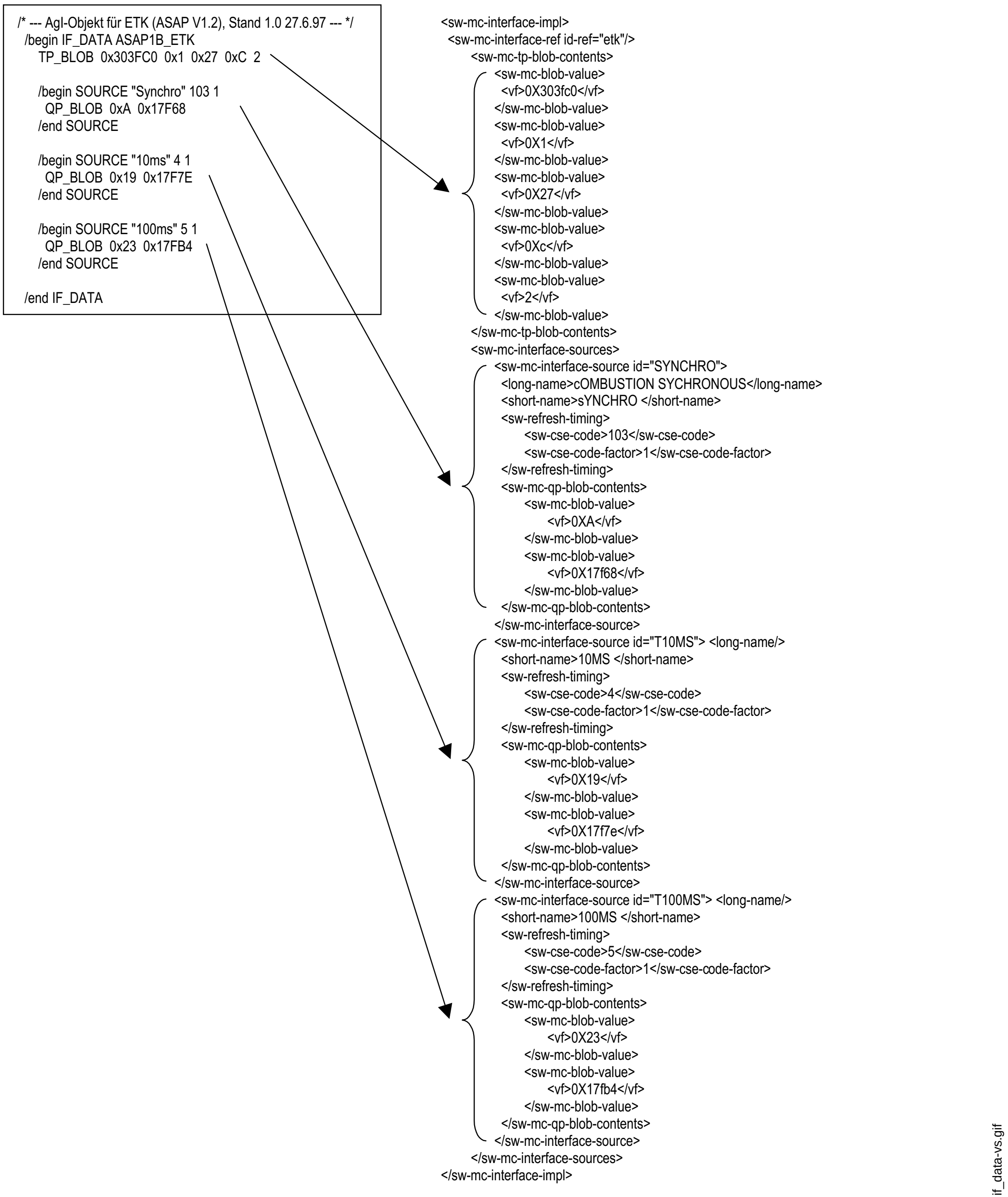



**Figure 16: IF-data**

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# 5 Annotated Sample file

This chapter shows an annotated sample file. The following hints apply:

- This file is kept simple for easy understanding. With this respect it does not show all capabilities and flexibility of the format, especially the new features such as classes and multiple instantiation.
- The file is primarily for an audience familiar with *ASAM-MCD-2MC V1.x* format. Other usecases such as system design, software documentation etc. are not covered here. Note that all references are made via the standard SGML mechanism (*ID/IDREF*) as well as by MSR natural addressing which refers to the short-name of the target object[10].
- The corresponding ASAM-MCD-2MC - file is displayed in Topic 5.4.4 Interface sources p. 54 for reference purposes.
- To some respect the correspondance is illustrated in
- The file is beautified using a standard SGML tool for readability.

## 5.1 The SGML document type declaration

The File starts with the SGML document type declaration. Note that there might also be a public identifier.

The **<msrsw>** container corresponds to a particular electronic control module, e.g. the engine control module.

The **<short-name>** identifies the file for reference purposes.

```
<!DOCTYPE MSRSW PUBLIC "-//MSR//DTD MSR SOFTWARE DTD:V2.2.0:MSRSW.DTD//EN">

<msrsw>
  <short-name>_70192</short-name>
```

## 5.2 Project data

Project data receives all information about the project itself, such as **<companies>**, **<team-members>** etc.

```
  <project-data>
    <project>
      <label>ASAP2-Example</label>

      <companies>
        <company role="MANUFACTURER"
                 id  ="MSRSW">
          <long-name>MSR Software Working Group</long-name>
          <short-name>msrsw</short-name>

          <team-members>
            <team-member id="OJ">
              <long-name>Herr Smith</long-name>
```

[10] We are sorry, in this example not all namespaces may be correct. Also the MSR references are treated not to be case sensitive

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
            <short-name>HS</short-name>
            <roles>
              <role><?xm-replace_text {ROLE}></role>
            </roles>
          </team-member>
        </team-members>
      </company>
    </companies>
  </project>
</project-data
```

# 5.3 The software systems

A software system (*SYSTEM*) corresponds to a particular electronic control module, e.g. the engine control module.

It would be possible to have information about more than one software system in the same file. Here we have only one **<sw-system>**.

```
<sw-systems>
  <sw-system id="DIM">
    <long-name>this is SW-SYSTEM DIM</long-name>
    <short-name>DIM</short-name>
```

## 5.3.1 The data dictionary

Here we have the global data dictionary. Note that also **<admin-data>** is here to indicate a specific revision of the data dictionary. This allows to reuse the same data dictionary across multiple projects and still keeping track of the revisions.

```
    <sw-data-dictionary-spec>
      <admin-data>
        <language>en</language>
        <lite-revisions>
          <lite-revision>
            <revision-label>1.65 & more</revision-label>
          </lite-revision>
        </lite-revisions>
      </admin-data>
      <sw-data-dictionary>
```

### 5.3.1.1 Measurement units

First of all we define all the measurement units **<sw-units>** we need in the systems. The particular units may be related to si-units. This allows some MCD-Systems to convert measurement units in to user selectable units.

Note that for example, "GRAD-C" is defined based on "KELVIN".

```
        <sw-units>
          <sw-unit id="KELVIN">
            <long-name>Kelvin</long-name>
            <short-name>kelvin</short-name>
            <sw-unit-display>K</sw-unit-display>
            <si-unit thermodynamic-temperature-expo="1"></si-unit>
          </sw-unit>

          <sw-unit id="GRAD-C">
            <long-name>Grad C</long-name>
            <short-name>Grad-C</short-name>
            <sw-unit-display>Grad C</sw-unit-display>
            <sw-unit-conversion-method>
              <sw-unit-gradient>1</sw-unit-gradient>
              <sw-unit-offset>-273.15</sw-unit-offset>
            </sw-unit-conversion-method>
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
    <sw-unit-ref id-ref="KELVIN">KELVIN</sw-unit-ref>
  </sw-unit>

  <sw-unit id="SEC">
    <long-name>Seconds</long-name>
    <short-name>sec</short-name>
    <sw-unit-display>s</sw-unit-display>
    <si-unit time-expo="1"></si-unit>
  </sw-unit>

  <sw-unit id="MSEC">
    <long-name>Milliseconds</long-name>
    <short-name>msec</short-name>
    <sw-unit-display>msec</sw-unit-display>
    <sw-unit-conversion-method>
      <sw-unit-gradient>0.001</sw-unit-gradient>
      <sw-unit-offset>0</sw-unit-offset>
    </sw-unit-conversion-method>
    <sw-unit-ref id-ref="SEC">SEC</sw-unit-ref>
  </sw-unit>

  <sw-unit id="HERTZ">
    <long-name>Hertz</long-name>
    <short-name>Hz</short-name>
    <sw-unit-display>Hz</sw-unit-display>
    <si-unit time-expo="-1"></si-unit>
  </sw-unit>

  <sw-unit id="UPM">
    <long-name>Rotates per minute</long-name>
    <short-name>Upm</short-name>
    <sw-unit-display>U/min</sw-unit-display>
    <sw-unit-conversion-method>
      <sw-unit-gradient> 0.01666666666667</sw-unit-gradient>
      <sw-unit-offset>0</sw-unit-offset>
    </sw-unit-conversion-method>
    <sw-unit-ref id-ref="HERTZ">HERTZ</sw-unit-ref>
  </sw-unit>

  <sw-unit id="NM">
    <long-name>Newton meter</long-name>
    <short-name>Nm</short-name>
    <sw-unit-display>Nm</sw-unit-display>
    <si-unit length-expo="2"
             mass-expo  ="1"></si-unit>
  </sw-unit>

  <sw-unit id="NO-UNIT">
    <long-name>No unit</long-name>
    <short-name>no-unit</short-name>
    <sw-unit-display>-</sw-unit-display>
    <si-unit></si-unit>
  </sw-unit>
</sw-units>
```

#### 5.3.1.2 Variables

Here we define the variables (known as *MEASUREMENT* in *ASAM-MCD-2MC 1.x*) which are used in the system.

```
<sw-variables>
  <sw-variable id="NMOT">
    <long-name>engine speed</long-name>
    <short-name>nmot</short-name>
    <sw-data-def-props>
      <sw-base-type-ref id-ref="UBYTE">UBYTE</sw-base-type-ref>
      <sw-compu-method-ref id-ref="NMOT-UB-Q40">NMOT-UB-Q40</sw-compu-method-ref>
      <sw-data-constr-ref id-ref="DC-NMOT">DC-NMOT</sw-data-constr-ref>
      <sw-display-format>%5.0</sw-display-format>
    </sw-data-def-props>
  </sw-variable>

  <sw-variable id="TEVFA-W-KGE">
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
    <long-name>this is SW-VARIABLE TEVFA-W-KGE</long-name>
    <short-name>tevfa_w_kge</short-name>
    <sw-data-def-props>
      <sw-base-type-ref id-ref="UWORD">UWORD</sw-base-type-ref>
      <sw-compu-method-ref id-ref="TINJECTION-SY">TINJECTION-SY</sw-compu-method-ref>
      <sw-data-constr-ref id-ref="DC-TEVFA-W-KGE">DC-TEVFA-W-KGE</sw-data-constr-ref>
      <sw-display-format>%5.0</sw-display-format>
    </sw-data-def-props>
  </sw-variable>

  <sw-variable id="B-2PH">
    <long-name>Condition for the second phase edge</long-name>
    <short-name>B_2ph</short-name>
    <sw-data-def-props>
      <sw-base-type-ref id-ref="UWORD">UWORD</sw-base-type-ref>
      <sw-compu-method-ref id-ref="B-TRUE">B-TRUE</sw-compu-method-ref>
      <sw-display-format>%13.11</sw-display-format>
    </sw-data-def-props>
  </sw-variable>
</sw-variables>
```

### 5.3.1.3 Calibration parameters

Here we have the calibration parameters (known as *CHARACTERISTIC* in *ASAM-MCD-2MC 1.x*). One can see that e.g. **<desc>** is optional and therefore not always given.

The parameter ABGMSHIGH is a simple VALUE.

```
<sw-calprms>
  <sw-calprm id="ABGMSIGH">
    <long-name>threshold for exh. temp. for
       wiring-interruption with Ri-diagnosis downstr. ca</long-name>
    <short-name>ABGMSIGH</short-name>
    <category>VALUE</category>
    <sw-data-def-props>
      <sw-compu-method-ref
             id-ref="TEMP-UB-Q5-O50">temp_ub_q5_o50</sw-compu-method-ref>
      <sw-data-constr-ref
             id-ref="DC-ABGMSIGH">DC-ABGMSIGH</sw-data-constr-ref>
      <sw-display-format>%5.0</sw-display-format>
      <sw-record-layout-ref
             id-ref="KWUB">KWUB</sw-record-layout-ref>
    </sw-data-def-props>
  </sw-calprm>
```

ANALUN as is *CURVE* with a reference to the axis points with *AXIS_PTS_REF*, and therefore the x-axis has to be defined as its own parameter.

```
  <sw-calprm id="ANALUN">
    <long-name>
       number of combustions for deactivation
       after detected misfire</long-name>
    <short-name>ANALUN</short-name>
    <category>CURVE</category>
    <sw-data-def-props>
     <sw-calibration-access>CALIBRATION</sw-calibration-access>
     <sw-calprm-axis-set>
        <sw-calprm-axis>
          <sw-axis-index>1</sw-axis-index>
          <sw-axis-grouped>
            <sw-axis-index>1</sw-axis-index>
            <sw-calprm-ref
                  id-ref="SNM08DMUB">SNM08DMUB</sw-calprm-ref>
          </sw-axis-grouped>
        </sw-calprm-axis>
      </sw-calprm-axis-set>
      <sw-compu-method-ref
            id-ref="DEZ">DEZ</sw-compu-method-ref>
      <sw-data-constr-ref
            id-ref="DC-ANALUN">DC-ANALUN</sw-data-constr-ref>
      <sw-display-format>%5.1</sw-display-format>
      <sw-interpolation-method>linear</sw-interpolation-method>
      <sw-record-layout-ref
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
        id-ref="SIMPLE-CURVE">SIMPLE-CURVE</sw-record-layout-ref>
  </sw-data-def-props>
</sw-calprm>

<sw-calprm id="SNM08DMUB">
  <long-name>this is SW-PARAM SNM08DMUB</long-name>
  <short-name>SNM08DMUB</short-name>
  <desc>
     This is the x-axis of the ANALUN. "Missing" stuff
     is fetched from the referenced variable nmot.
     If needed the properties of nmot could be override.</desc>
  <category>COM_AXIS</category>
  <sw-data-def-props>
    <sw-calprm-axis-set>
      <sw-calprm-axis>
        <sw-axis-index>1</sw-axis-index>
        <sw-axis-individual>
          <sw-variable-refs>
            <sw-variable-ref id-ref="NMOT">NMOT</sw-variable-ref>
          </sw-variable-refs>
          <sw-max-axis-points>8</sw-max-axis-points>
        </sw-axis-individual>
      </sw-calprm-axis>
    </sw-calprm-axis-set>
    <sw-record-layout-ref id-ref="COMMON-X-AXIS">COMMON-X-AXIS</sw-record-layout-ref>
  </sw-data-def-props>
</sw-calprm>
```

Finally, the FKKVS is a *MAP* with embedded-x and y-axis.

```
<sw-calprm id="FKKVS">
  <long-name>factor to correct fuel delivery system</long-name>
  <short-name>FKKVS</short-name>
  <desc></desc>
  <category>MAP</category>
  <sw-data-def-props>
    <sw-calibration-access>CALIBRATION</sw-calibration-access>
    <sw-calprm-axis-set>
      <sw-calprm-axis>
        <sw-axis-index>1</sw-axis-index>
        <sw-axis-individual>
          <sw-variable-refs>
            <sw-variable-ref id-ref="NMOT">NMOT</sw-variable-ref>
          </sw-variable-refs>
          <sw-max-axis-points>16</sw-max-axis-points>
        </sw-axis-individual>
      </sw-calprm-axis>
      <sw-calprm-axis>
        <sw-axis-index>2</sw-axis-index>
        <sw-axis-individual>
          <sw-variable-refs>
            <sw-variable-ref id-ref="TEVFA-W-KGE">TEVFA-W-KGE</sw-variable-ref>
          </sw-variable-refs>
          <sw-max-axis-points>16</sw-max-axis-points>
        </sw-axis-individual>
      </sw-calprm-axis>
    </sw-calprm-axis-set>
    <sw-compu-method-ref id-ref="FAK-UW-B2">FAK-UW-B2</sw-compu-method-ref>
    <sw-data-constr-ref id-ref="DC-FKKVS">DC-FKKVS</sw-data-constr-ref>
    <sw-display-format>%8.6</sw-display-format>
    <sw-interpolation-method>linear</sw-interpolation-method>
    <sw-record-layout-ref id-ref="SIMPLE-MAP">SIMPLE-MAP</sw-record-layout-ref>
  </sw-data-def-props>
</sw-calprm>
</sw-calprms>
```

It is possible to define Parameters with more than three axis (today's *CUBOID*), but it will be a copletetely other problem to present it for the user.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

### 5.3.1.4 System constants

System constants are defined in order to reconfigure the system by changing the value of particular system constants. Note that these system constants are given es example but are not referenced in this sample file.

```
<sw-systemconsts>
  <sw-systemconst id="EPK-MAX-LEN">
    <long-name>this is SW-SYSTEM-CONSTANT EPK-MAX-LEN</long-name>
    <short-name>EPK_MAX_LEN</short-name>
    <sw-values-phys><v>80</v></sw-values-phys>
  </sw-systemconst>

  <sw-systemconst id="SWOFFDL-REQUEST-MASK">
    <long-name>this is SW-SYSTEM-CONSTANT SWOFFDL-REQUEST-MASK</long-name>
    <short-name>SWOFFDL_REQUEST_MASK</short-name>
    <sw-values-phys><v>16383</v></sw-values-phys>
  </sw-systemconst>

  <sw-systemconst id="SY-2SG">
    <long-name>this is SW-SYSTEM-CONSTANT SY-2SG</long-name>
    <short-name>SY_2SG</short-name>
    <sw-values-phys><v>0</v></sw-values-phys>
  </sw-systemconst>
</sw-systemconsts>
```

### 5.3.1.5 Computation methods

Computation methods indicate how the physical world corresponds to the normalized domain.

```
<sw-compu-methods>
  <sw-compu-method id="TEMP-UB-Q5-O50">
    <long-name>this is SW-COMPU-METHOD TEMP-UB-Q5-O50</long-name>
    <short-name>temp_ub_q5_o50</short-name>
    <sw-display-format>%6.2</sw-display-format>
    <sw-unit-ref id-ref="GRAD-C">GRAD-C</sw-unit-ref>
    <sw-compu-phys-to-internal>
      <sw-compu-scales>
        <sw-compu-scale>
          <sw-compu-inverse-value>
            <v>1</v>
          </sw-compu-inverse-value>
          <sw-compu-rational-coeffs>
            <sw-compu-numerator>
              <vf>50</vf>
              <vf>1</vf>
            </sw-compu-numerator>
            <sw-compu-denominator>
              <vf>5</vf>
            </sw-compu-denominator>
          </sw-compu-rational-coeffs>
        </sw-compu-scale>
      </sw-compu-scales>
    </sw-compu-phys-to-internal>
  </sw-compu-method>

  <sw-compu-method id="DEZ">
    <long-name>1 : 1 conversion (dez/hex)</long-name>
    <short-name>dez</short-name>
    <sw-display-format>%6.2</sw-display-format>
    <sw-compu-identity></sw-compu-identity>
  </sw-compu-method>
```

The six parameter formula (*RAT_FUNC*) has been removed and substituted by a more generic solution which uses a numerator (**<sw-compu-numerator>**) and a denominator **<sw-compu-denominator>**. Each of this is a polynomium for which the factors are specified (**<vf>**) starting with the factor for $x^0$. Therefore the example defines:

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

### Example for rational function

$$internal = \frac{phys}{40}$$

```
<sw-compu-method id=”NMOT-UB-Q40”>
  <long-name>this is SW-COMPU-METHOD NMOT-UB-Q40</long-name>
  <short-name>nmot_ub_q40</short-name>
  <sw-display-format>%6.1</sw-display-format>
  <sw-unit-ref id-ref=”GRAD-C”>GRAD-C</sw-unit-ref>
  <sw-compu-phys-to-internal>
    <sw-compu-scales>
      <sw-compu-scale>
        <sw-compu-inverse-value>
          <v>1</v>
        </sw-compu-inverse-value>
        <sw-compu-rational-coeffs>
          <sw-compu-numerator>
            <vf>0</vf>
            <vf>1</vf>
          </sw-compu-numerator>
          <sw-compu-denominator>
            <vf>40</vf>
          </sw-compu-denominator>
        </sw-compu-rational-coeffs>
      </sw-compu-scale>
    </sw-compu-scales>
  </sw-compu-phys-to-internal>
</sw-compu-method>

<sw-compu-method id=”B-TRUE”>
  <long-name>this is SW-COMPU-METHOD B-TRUE</long-name>
  <short-name>B_TRUE</short-name>
  <sw-display-format>%6.3</sw-display-format>
  <sw-compu-internal2phys>
    <sw-compu-scales>
      <sw-compu-scale>
        <lower-limit interval-type=”CLOSED”>0</lower-limit>
        <upper-limit interval-type=”CLOSED”>0</upper-limit>
        <sw-compu-inverse-value>
          <v>0</v>
        </sw-compu-inverse-value>
        <sw-compu-const>
          <vt>—-</vt>
        </sw-compu-const>
      </sw-compu-scale>
      <sw-compu-scale>
        <lower-limit interval-type=”CLOSED”>1</lower-limit>
        <upper-limit interval-type=”CLOSED”>1</upper-limit>
        <sw-compu-inverse-value>
          <v>1</v>
        </sw-compu-inverse-value>
        <sw-compu-const>
          <vt>TRUE</vt>
        </sw-compu-const>
      </sw-compu-scale>
      <sw-compu-scale>
        <lower-limit interval-type=”CLOSED”>2</lower-limit>
        <upper-limit interval-type=”CLOSED”>2</upper-limit>
        <sw-compu-inverse-value>
          <v>2</v>
        </sw-compu-inverse-value>
        <sw-compu-const>
          <vt></vt>
        </sw-compu-const>
      </sw-compu-scale>
    </sw-compu-scales>
  </sw-compu-internal2phys>
</sw-compu-method>

<sw-compu-method id=”TINJECTION-SY”>
  <long-name>this is SW-COMPU-METHOD TINJECTION-SY</long-name>
  <short-name>tinjection_sy</short-name>
  <sw-display-format>%6.4</sw-display-format>
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
    <sw-unit-ref id-ref="MSEC">MSEC</sw-unit-ref>
    <sw-compu-phys-to-internal>
      <sw-compu-scales>
        <sw-compu-scale>
          <sw-compu-inverse-value>
            <v>1</v>
          </sw-compu-inverse-value>
          <sw-compu-rational-coeffs>
            <sw-compu-numerator>
              <vf>0</vf>
              <vf>500</vf>
            </sw-compu-numerator>
            <sw-compu-denominator>
              <vf>1</vf>
            </sw-compu-denominator>
          </sw-compu-rational-coeffs>
        </sw-compu-scale>
      </sw-compu-scales>
    </sw-compu-phys-to-internal>
  </sw-compu-method>

  <sw-compu-method id="FAK-UW-B2">
    <long-name>this is SW-COMPU-METHOD FAK-UW-B2</long-name>
    <short-name>fak_uw_b2</short-name>
    <sw-display-format>%7.6</sw-display-format>
    <sw-unit-ref id-ref="NO-UNIT">NO-UNIT</sw-unit-ref>
    <sw-compu-phys-to-internal>
      <sw-compu-scales>
        <sw-compu-scale>
          <sw-compu-inverse-value>
            <v>1</v>
          </sw-compu-inverse-value>
          <sw-compu-rational-coeffs>
            <sw-compu-numerator>
              <vf>0</vf>
              <vf>65536</vf>
            </sw-compu-numerator>
            <sw-compu-denominator>
              <vf>2</vf>
            </sw-compu-denominator>
          </sw-compu-rational-coeffs>
        </sw-compu-scale>
      </sw-compu-scales>
    </sw-compu-phys-to-internal>
  </sw-compu-method>
</sw-compu-methods>
```

### 5.3.1.6 Record layouts

The record layouts provide a scheme to indicate how the objects must be stored in memory. The sample file shows more record layouts than really needed in order to illustrate the flexibility of the concept.

Note the iterator indexes count in reverse order if they are negative. Therefore value ”-1” denotes the last item of a set, ”-3” denotes the third last item of a set etc.

Iterations are setup using **<sw-record-layout-group>** where the attributes indicate the properties of the iteration.

```
<sw-record-layouts>
  <sw-record-layout id="KWUB">
    <long-name>this is SW-RECORD-LAYOUT KWUB</long-name>
    <short-name>KwUb</short-name>
    <sw-record-layout-group>
      <sw-record-layout-group-from>1</sw-record-layout-group-from>
      <sw-record-layout-group-to>-1</sw-record-layout-group-to>
      <sw-record-layout-v>
        <sw-base-type-ref id-ref="UBYTE">UBYTE</sw-base-type-ref>
        <sw-record-layout-v-prop>VALUE</sw-record-layout-v-prop>
      </sw-record-layout-v>
    </sw-record-layout-group>
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
</sw-record-layout>
```

#### 5.3.1.6.1 Record layout for simple curve

In the simple curve after the size of the curve and the source adress the x-axis then the value axis is stored.

```
<sw-record-layout id="SIMPLE-CURVE">
  <long-name>This is the record layout for a simple curve.</long-name>
  <short-name>rly-simple-curve</short-name>
  <sw-record-layout-group>
    <sw-record-layout-v>
      <sw-base-type-ref id-ref="UBYTE">UBYTE</sw-base-type-ref>
      <sw-record-layout-v-prop>SOURCE-ADR</sw-record-layout-v-prop>
    </sw-record-layout-v>
    <sw-record-layout-v>
      <sw-record-layout-v-prop>COUNT</sw-record-layout-v-prop>
    </sw-record-layout-v>
```

This group iterates on axis number 1 (which is the x-axis). The iterator has the symbolic name ”x” and runs from the first to the last axis point. The group handles the axis points.

```
    <sw-record-layout-group>
      <desc>Here are the values of the x axis.</desc>
      <sw-record-layout-group-axis>1</sw-record-layout-group-axis>
      <sw-record-layout-group-index>x</sw-record-layout-group-index>
      <sw-record-layout-group-from>1</sw-record-layout-group-from>
      <sw-record-layout-group-to>-1</sw-record-layout-group-to>
      <sw-record-layout-v>
        <sw-record-layout-v-prop>VALUE</sw-record-layout-v-prop>
        <sw-record-layout-v-index>x</sw-record-layout-v-index>
      </sw-record-layout-v>
    </sw-record-layout-group>
```

This group iterates on axis number 1 (which is the x-axis). The iterator has the symbolic name ”x” and runs from the first to the last axis point. The group handles the value points.

```
    <sw-record-layout-group>
      <desc>Here are the values of the value axis.</desc>
      <sw-record-layout-group-axis>0</sw-record-layout-group-axis>
      <sw-record-layout-group-index>v</sw-record-layout-group-index>
      <sw-record-layout-group-from>1</sw-record-layout-group-from>
      <sw-record-layout-group-to>-1</sw-record-layout-group-to>
      <sw-record-layout-v>
        <sw-record-layout-v-prop>VALUE</sw-record-layout-v-prop>
        <sw-record-layout-v-index>v</sw-record-layout-v-index>
      </sw-record-layout-v>
    </sw-record-layout-group>
  </sw-record-layout-group>
</sw-record-layout>
```

#### 5.3.1.6.2 Record layout for alternative curve

In the alternative curve, the iterator runs through the x-axis but both, the axis point and the corresponding value are handled.

```
<sw-record-layout id="ALTERNATIVE-CURVE">
  <long-name>This is the record layout for an alternative curve.</long-name>
  <short-name>alternative-curve</short-name>
  <sw-record-layout-group>
    <sw-record-layout-v>
      <sw-base-type-ref id-ref="UBYTE">UBYTE</sw-base-type-ref>
      <sw-record-layout-v-prop>SOURCE-ADR</sw-record-layout-v-prop>
    </sw-record-layout-v>
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
    <sw-record-layout-v>
      <sw-record-layout-v-prop>COUNT</sw-record-layout-v-prop>
    </sw-record-layout-v>
    <sw-record-layout-group>
      <desc>Here are the values of the x axis.</desc>
      <sw-record-layout-group-axis>1</sw-record-layout-group-axis>
      <sw-record-layout-group-index>x</sw-record-layout-group-index>
      <sw-record-layout-group-from>1</sw-record-layout-group-from>
      <sw-record-layout-group-to>-1</sw-record-layout-group-to>
```

Based on this, the axis point and related value are placed adjacently in memory.

```
      <sw-record-layout-v>
        <sw-record-layout-v-axis>1</sw-record-layout-v-axis>
        <sw-record-layout-v-prop>VALUE</sw-record-layout-v-prop>
        <sw-record-layout-v-index>x</sw-record-layout-v-index>
      </sw-record-layout-v>
      <sw-record-layout-v>
        <sw-record-layout-v-axis>0</sw-record-layout-v-axis>
        <sw-record-layout-v-prop>VALUE</sw-record-layout-v-prop>
        <sw-record-layout-v-index>x</sw-record-layout-v-index>
      </sw-record-layout-v>
    </sw-record-layout-group>
  </sw-record-layout-group>
</sw-record-layout>
```

#### 5.3.1.6.3 Record layout for simple map

In the simple map, we first place the x-axis, then the y-axis.

```
<sw-record-layout id=”SIMPLE-MAP”>
  <long-name>This is the record layout for a simple map.</long-name>
  <short-name>simple-map</short-name>
  <desc>Appendix B in ASAP2-specification (KEBUSS).</desc>
  <sw-record-layout-group>
    <sw-record-layout-group>
      <sw-record-layout-v>
        <sw-base-type-ref id-ref=”UBYTE”>UBYTE</sw-base-type-ref>
        <sw-record-layout-v-axis>1</sw-record-layout-v-axis>
        <sw-record-layout-v-prop>COUNT</sw-record-layout-v-prop>
      </sw-record-layout-v>
      <sw-record-layout-v>
        <sw-record-layout-v-axis>1</sw-record-layout-v-axis>
        <sw-record-layout-v-prop>COUNT</sw-record-layout-v-prop>
      </sw-record-layout-v>
    </sw-record-layout-group>
    <sw-record-layout-group>
      <desc>Here are the values of the x axis.</desc>
      <sw-record-layout-group-axis>1</sw-record-layout-group-axis>
      <sw-record-layout-group-index>x</sw-record-layout-group-index>
      <sw-record-layout-group-from>-1</sw-record-layout-group-from>
      <sw-record-layout-group-to>1</sw-record-layout-group-to>
      <sw-record-layout-v>
        <sw-base-type-ref id-ref=”UBYTE”>UBYTE</sw-base-type-ref>
        <sw-record-layout-v-axis>1</sw-record-layout-v-axis>
        <sw-record-layout-v-prop>VALUE</sw-record-layout-v-prop>
        <sw-record-layout-v-index>x</sw-record-layout-v-index>
      </sw-record-layout-v>
    </sw-record-layout-group>
    <sw-record-layout-group>
      <desc>Here are the values of the y axis.</desc>
      <sw-record-layout-group-axis>2</sw-record-layout-group-axis>
      <sw-record-layout-group-index>y</sw-record-layout-group-index>
      <sw-record-layout-group-from>-1</sw-record-layout-group-from>
      <sw-record-layout-group-to>1</sw-record-layout-group-to>
      <sw-record-layout-v>
        <sw-base-type-ref id-ref=”UWORD”>UWORD</sw-base-type-ref>
        <sw-record-layout-v-axis>2</sw-record-layout-v-axis>
        <sw-record-layout-v-prop>VALUE</sw-record-layout-v-prop>
        <sw-record-layout-v-index>x</sw-record-layout-v-index>
      </sw-record-layout-v>
    </sw-record-layout-group>
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

To place the value map, two nested record layout groups are used. The inner one (for the x-axis) runs faster.

```
        <sw-record-layout-group>
          <sw-record-layout-group-axis>2</sw-record-layout-group-axis>
          <sw-record-layout-group-index>y</sw-record-layout-group-index>
          <sw-record-layout-group-from>-1</sw-record-layout-group-from>
          <sw-record-layout-group-to>1</sw-record-layout-group-to>
          <sw-record-layout-group>
            <sw-record-layout-group-axis>1</sw-record-layout-group-axis>
            <sw-record-layout-group-index>x</sw-record-layout-group-index>
            <sw-record-layout-group-from>-1</sw-record-layout-group-from>
            <sw-record-layout-group-to>1</sw-record-layout-group-to>
            <sw-record-layout-v>
              <sw-base-type-ref id-ref="UWORD">UWORD</sw-base-type-ref>
              <sw-record-layout-v-axis>0</sw-record-layout-v-axis>
              <sw-record-layout-v-prop>VALUE</sw-record-layout-v-prop>
              <sw-record-layout-v-index>x y</sw-record-layout-v-index>
            </sw-record-layout-v>
          </sw-record-layout-group>
        </sw-record-layout-group>
      </sw-record-layout-group>
    </sw-record-layout>
```

## 5.3.1.6.4 Record layout for common axis

```
    <sw-record-layout id="COMMON-X-AXIS">
      <long-name>This is the record layout for a common x Axis.</long-name>
      <short-name>common-x-axis</short-name>
      <desc></desc>
      <sw-record-layout-group>
        <sw-record-layout-group>
          <sw-record-layout-v>
            <sw-base-type-ref id-ref="UBYTE">UBYTE</sw-base-type-ref>
            <sw-record-layout-v-axis>1</sw-record-layout-v-axis>
            <sw-record-layout-v-prop>COUNT</sw-record-layout-v-prop>
          </sw-record-layout-v>
        </sw-record-layout-group>
        <sw-record-layout-group>
          <desc>Here are the values of the x axis.</desc>
          <sw-record-layout-group-axis>1</sw-record-layout-group-axis>
          <sw-record-layout-group-index>x</sw-record-layout-group-index>
          <sw-record-layout-group-from>1</sw-record-layout-group-from>
          <sw-record-layout-group-to>-1</sw-record-layout-group-to>
          <sw-record-layout-v>
            <sw-base-type-ref id-ref="UBYTE">UBYTE</sw-base-type-ref>
            <sw-record-layout-v-axis>1</sw-record-layout-v-axis>
            <sw-record-layout-v-prop>VALUE</sw-record-layout-v-prop>
            <sw-record-layout-v-index>x</sw-record-layout-v-index>
          </sw-record-layout-v>
        </sw-record-layout-group>
      </sw-record-layout-group>
    </sw-record-layou
```

## 5.3.1.7 Base types for the CPU

Here the base types of the CPU are defined.

```
  <sw-base-types>
    <sw-base-type id="UBYTE">
      <long-name>Unsigned byte</long-name>
      <short-name>ubyte</short-name>
      <sw-base-type-size>8</sw-base-type-size>
      <sw-coded-type>unsigned</sw-coded-type>
      <sw-mem-alignment>8</sw-mem-alignment>
      <byte-order type="MOST-SIGNIFICANT-BYTE-LAST"></byte-order>
    </sw-base-type>

    <sw-base-type id="UWORD">
      <long-name>Unsigned word</long-name>
      <short-name>uword</short-name>
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
    <sw-base-type-size>16</sw-base-type-size>
    <sw-coded-type>unsigned</sw-coded-type>
    <sw-mem-alignment>16</sw-mem-alignment>
    <byte-order type="MOST-SIGNIFICANT-BYTE-LAST"></byte-order>
  </sw-base-type>
</sw-base-types>
```

### 5.3.1.8 Constraints

Constraints are defined here and may be applied to any data object. Note that even if it appears the a separate constraint object is defined for each variable or parameter this is not realy the case. Constraints may be reused wherever it is appropriate.

```
<sw-data-constrs>
 <sw-data-constr id="DC-NMOT">
   <long-name>this is SW-DATA-CONSTRAINT DC-NMOT</long-name>
   <short-name>nmot</short-name>
   <sw-data-constr-rule>
     <sw-constr-level></sw-constr-level>
     <sw-phys-constrs>
       <sw-scale-constrs>
         <sw-scale-constr>
           <lower-limit interval-type="CLOSED">0</lower-limit>
           <upper-limit interval-type="CLOSED">10200</upper-limit>
         </sw-scale-constr>
       </sw-scale-constrs>
     </sw-phys-constrs>
   </sw-data-constr-rule>
 </sw-data-constr>

 <sw-data-constr id="DC-ABGMSIGH">
   <long-name>this is SW-DATA-CONSTRAINT DC-ABGMSIGH</long-name>
   <short-name>ABGMSIGH</short-name>
   <sw-data-constr-rule>
     <sw-constr-level>0</sw-constr-level>
     <sw-phys-constrs>
       <sw-scale-constrs>
         <sw-scale-constr>
           <lower-limit interval-type="CLOSED">-50.00</lower-limit>
           <upper-limit interval-type="CLOSED">1225</upper-limit>
         </sw-scale-constr>
       </sw-scale-constrs>
     </sw-phys-constrs>
   </sw-data-constr-rule>
 </sw-data-constr>

 <sw-data-constr id="DC-ANALUN">
   <long-name>this is SW-DATA-CONSTRAINT DC-ANALUN</long-name>
   <short-name>ANALUN</short-name>
   <category></category>
   <sw-data-constr-rule>
     <sw-constr-level>0</sw-constr-level>
     <sw-phys-constrs>
       <sw-scale-constrs>
         <sw-scale-constr>
           <lower-limit interval-type="CLOSED">0.0</lower-limit>
           <upper-limit interval-type="CLOSED">255.0</upper-limit>
         </sw-scale-constr>
       </sw-scale-constrs>
     </sw-phys-constrs>
   </sw-data-constr-rule>
 </sw-data-constr>

 <sw-data-constr id="DC-TEVFA-W-KGE">
   <long-name>this is SW-DATA-CONSTRAINT DC-TEVFA-W-KGE</long-name>
   <short-name>tevfa_w_kge</short-name>
   <sw-data-constr-rule>
     <sw-constr-level>0</sw-constr-level>
     <sw-phys-constrs>
       <sw-scale-constrs>
         <sw-scale-constr>
           <lower-limit interval-type="CLOSED">0.0</lower-limit>
           <upper-limit interval-type="CLOSED">131.07</upper-limit>
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
            </sw-scale-constr>
          </sw-scale-constrs>
        </sw-phys-constrs>
      </sw-data-constr-rule>
    </sw-data-constr>

    <sw-data-constr id="DC-FKKVS">
      <long-name>this is SW-DATA-CONSTRAINT DC-FKKVS</long-name>
      <short-name>FKKVS</short-name>
      <category></category>
      <sw-data-constr-rule>
        <sw-constr-level>0</sw-constr-level>
        <sw-phys-constrs>
          <sw-scale-constrs>
            <sw-scale-constr>
              <lower-limit interval-type="CLOSED">0.0</lower-limit>
              <upper-limit interval-type="CLOSED">1.999969</upper-limit>
            </sw-scale-constr>
          </sw-scale-constrs>
        </sw-phys-constrs>
      </sw-data-constr-rule>
    </sw-data-constr>
  </sw-data-constrs>
</sw-data-dictionary-spec>
```

## 5.3.2 The instances of variables and parameters

Here we have the instances of variables and parameters. This part was **not** really there in *ASAM-MCD-2MC V1.x* files.

```
<sw-instance-spec>
  <sw-instance-tree>
    <short-name>_70192</short-name>
```

Here we have a special case: the **<short-name>** acts as reference as well as as identifier. Through the role of reference the link to the data dictionary is established. Note that **<long-name>** is somewhat redundant here. The DTD does not require it. But for sake of clarity, we repeat the long-name here. It also shows that the long-name may be specified instance-specific (of whatever use this may be).

The instance could contain, in principle, only the changes between two (or more) compilations whcih imply that **<sw-data-dictionary>** is not changed. In this sample only the addresses and BLOBS are given in the instance.

```
<sw-instance id="IT-NMOT">
  <long-name>engine-speed (measurement)</long-name>
  <short-name>nmot</short-name>
  <sw-instance-props-variants>
    <sw-instance-props-variant>
      <sw-addr-infos>
        <sw-addr-info>
          <sw-base-addr>0xF86C</sw-base-addr>
        </sw-addr-info>
      </sw-addr-infos>
```

Each variable and parameter has a reference to the BLOB specification (**<sw-mc-interface-ref>**). It is the implemented in between e.g. **<sw-mc-kp-blob-contents>**.

```
 <sw-instance id="IT-B-2PH">
  <long-name>Condition for the second phase edge (measurement)</long-name>
  <short-name>B_2ph</short-name>
  <sw-instance-props-variants>
    <sw-instance-props-variant>
      <sw-addr-infos>
        <sw-addr-info>
          <sw-base-addr>0xFD0C</sw-base-addr>
        </sw-addr-info>
      </sw-addr-infos>
      <sw-mc-instance-interfaces>
        <sw-mc-instance-interface>
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
          <sw-mc-interface-ref id-ref=”ETK”>ETK</sw-mc-interface-ref>
          <sw-mc-kp-blob-conts>
            0xFD0C 0x0 0x2
          </sw-mc-kp-blob-conts>
        </sw-mc-instance-interface>
        <sw-mc-instance-interface>
          <sw-mc-interface-ref id-ref=”ADDRESS”>ADDRESS</sw-mc-interface-ref>
          <sw-mc-kp-blob-conts>
            0xFD0C
          </sw-mc-kp-blob-conts>
        </sw-mc-instance-interface>
      </sw-mc-instance-interfaces>
    </sw-instance-props-variant>
  </sw-instance-props-variants>
</sw-instance>

<sw-instance id=”IT-ABGMSIGH”>
  <long-name>threshold for exh. temp. for wiring-interruption with Ri-diagnosis downstr. cat (s
  <short-name>ABGMSIGH</short-name>
  <sw-instance-props-variants>
    <sw-instance-props-variant>
      <sw-addr-infos>
        <sw-addr-info>
          <sw-base-addr>0x1A0A8</sw-base-addr>
        </sw-addr-info>
      </sw-addr-infos>
      <sw-mc-instance-interfaces>
        <sw-mc-instance-interface>
          <sw-mc-interface-ref id-ref=”ETK”>ETK</sw-mc-interface-ref>
          <sw-mc-dp-blob-conts>
            0x1A0A8 0x1
          </sw-mc-dp-blob-conts>
        </sw-mc-instance-interface>
      </sw-mc-instance-interfaces>
    </sw-instance-props-variant>
  </sw-instance-props-variants>
</sw-instance>

<sw-instance id=”IT-ANALUN”>
  <long-name>(curve)</long-name>
  <short-name>ANALUN</short-name>
  <sw-instance-props-variants>
    <sw-instance-props-variant>
      <sw-addr-infos>
        <sw-addr-info>
          <sw-base-addr>0x1125C</sw-base-addr>
        </sw-addr-info>
      </sw-addr-infos>
      <sw-mc-instance-interfaces>
        <sw-mc-instance-interface>
          <sw-mc-interface-ref id-ref=”ETK”>ETK</sw-mc-interface-ref>
          <sw-mc-dp-blob-conts>
            0x1125C 0x8
          </sw-mc-dp-blob-conts>
        </sw-mc-instance-interface>
      </sw-mc-instance-interfaces>
    </sw-instance-props-variant>
  </sw-instance-props-variants>
</sw-instance>

<sw-instance id=”IT-TEVFA-W-KGE”>
  <long-name>(measurement)</long-name>
  <short-name>tevfa_w_kge</short-name>
  <sw-instance-props-variants>
    <sw-instance-props-variant>
      <sw-addr-infos>
        <sw-addr-info>
          <sw-base-addr>0x3012A2</sw-base-addr>
        </sw-addr-info>
      </sw-addr-infos>
      <sw-mc-instance-interfaces>
        <sw-mc-instance-interface>
          <sw-mc-interface-ref id-ref=”ETK”>ETK</sw-mc-interface-ref>
          <sw-mc-kp-blob-conts>
            0x3012A2 0x1 0x2
          </sw-mc-kp-blob-conts>
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
          </sw-mc-instance-interface>
          <sw-mc-instance-interface>
            <sw-mc-interface-ref id-ref="ADDRESS">ADDRESS</sw-mc-interface-ref>
            <sw-mc-kp-blob-conts>
              0x3012A2
            </sw-mc-kp-blob-conts>
          </sw-mc-instance-interface>
        </sw-mc-instance-interfaces>
      </sw-instance-props-variant>
    </sw-instance-props-variants>
  </sw-instance>

  <sw-instance id="IT-SNM08DMUB">
    <long-name>(x axis)</long-name>
    <short-name>SNM08DMUB</short-name>
    <sw-instance-props-variants>
      <sw-instance-props-variant>
        <sw-addr-infos>
          <sw-addr-info>
            <sw-base-addr>0xD918</sw-base-addr>
          </sw-addr-info>
        </sw-addr-infos>
        <sw-mc-instance-interfaces>
          <sw-mc-instance-interface>
            <sw-mc-interface-ref id-ref="ETK">ETK</sw-mc-interface-ref>
            <sw-mc-dp-blob-conts>
              0xD918 0x9
            </sw-mc-dp-blob-conts>
          </sw-mc-instance-interface>
        </sw-mc-instance-interfaces>
      </sw-instance-props-variant>
    </sw-instance-props-variants>
  </sw-instance>

  <sw-instance id="IT-FKKVS">
    <long-name>factor to correct fuel delivery system (map)</long-name>
    <short-name>FKKVS</short-name>
    <sw-instance-props-variants>
      <sw-instance-props-variant>
        <sw-addr-infos>
          <sw-addr-info>
            <sw-base-addr>0x1FA40</sw-base-addr>
          </sw-addr-info>
        </sw-addr-infos>
        <sw-mc-instance-interfaces>
          <sw-mc-instance-interface>
            <sw-mc-interface-ref id-ref="ETK">ETK</sw-mc-interface-ref>
            <sw-mc-dp-blob-conts>
              0x1FA40 0x232
            </sw-mc-dp-blob-conts>
          </sw-mc-instance-interface>
        </sw-mc-instance-interfaces>
      </sw-instance-props-variant>
    </sw-instance-props-variants>
  </sw-instance>
 </sw-instance-tree>
</sw-instance-spec>
```

### 5.3.3 Specification of the CPU

This section receives the CPU related information.

```
<sw-cpu-mem-segs>
  <sw-cpu-mem-seg id="MEM-SEG-1">
    <short-name>mem-seg-1</short-name>
    <sw-mem-program-type>CODE</sw-mem-program-type>
    <sw-mem-type>rom</sw-mem-type>
    <sw-mem-attr>intern</sw-mem-attr>
    <sw-mem-base-addr>0x0</sw-mem-base-addr>
    <sw-mem-size>0xc000</sw-mem-size>
    <sw-mc-instance-interfaces>
      <sw-mc-instance-interface>
        <sw-mc-interface-ref id-ref="ETK">ETK</sw-mc-interface-ref>
        <sw-mc-dp-blob-conts>
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
          0x0 0xc000
        </sw-mc-dp-blob-conts>
      </sw-mc-instance-interface>
    </sw-mc-instance-interfaces>
  </sw-cpu-mem-seg>

  <sw-cpu-mem-seg id="MEM-SEG-2">
    <short-name>mem-seg-2</short-name>
    <sw-mem-program-type>DATA</sw-mem-program-type>
    <sw-mem-type>ram</sw-mem-type>
    <sw-mem-attr>intern</sw-mem-attr>
    <sw-mem-base-addr>0xc000</sw-mem-base-addr>
    <sw-mem-size>0x2000</sw-mem-size>
    <sw-mc-instance-interfaces>
      <sw-mc-instance-interface>
        <sw-mc-interface-ref id-ref="ETK">ETK</sw-mc-interface-ref>
        <sw-mc-dp-blob-conts>
          0xc000 0x2000
        </sw-mc-dp-blob-conts>
      </sw-mc-instance-interface>
    </sw-mc-instance-interfaces>
  </sw-cpu-mem-seg>
</sw-cpu-mem-segs>
<sw-cpu-epk></sw-cpu-epk>
<sw-cpu-addr-epk>
  <sw-addr-info>
    <sw-base-addr>0xFFFF</sw-base-addr>
  </sw-addr-info>
</sw-cpu-addr-epk>
<sw-cpu-type>S80166</sw-cpu-type>
</sw-cpu-spec>
</sw-system>
```

Here we are also at the end of the software system.

## 5.4 Communication with MC-Systems

This section specifies the communication with the *MCD-System* (known as the *AML* part in the *ASAM-MCD-2MC 1.x*. The **<msrsw>** container corresponds to a particular electronic control module, eg. the engine control module. Since this module has one interface, the **<sw-mc-communication-spec>** is common across a all systems within the module.

```
<sw-mc-communication-spec>
```

### 5.4.1 Interface specification for MC systems

The interface specification merely consists of the BLOB layouts. For each interface and BLOB (TP, QP, KP, DP and PA) a aprticular layout is defined (e.g. **<sw-mc-kp-blob-layout>** ).

```
<sw-mc-communication-spec>
  <sw-mc-interface-spec>
    <sw-mc-interface id="ETK">
      <short-name>ETK</short-name>
      <sw-mc-blob-layouts>
        <sw-mc-tp-blob-layout>
          long; /* Trigger-Segment-Adresse */
          int;  /* Ausgabeformat der Displaytabelle 1=byte 2=word */
          int;  /* Trigger-Modus (spezielle Codierung) */
          int;  /* Typ der Displaytabelle (12, 11, 20) */
          int;  /* Byte-Reihenfolge 1 = high first, 2 = low first */
        </sw-mc-tp-blob-layout>
        <sw-mc-qp-blob-layout>
          int;  /* Max. Groesse Displaytabelle */
          long; /* Adresse Displaytabelle */
        </sw-mc-qp-blob-layout>
        <sw-mc-kp-blob-layout>
          long; /* Anfangs-Adresse */
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
        int;  /* Adress-Ort (0=intern, 1=extern) */
        int;  /* Anzahl der Bytes */
      </sw-mc-kp-blob-layout>
      <sw-mc-dp-blob-layout>
        long; /* Anfangs-Adresse */
        long; /* Laenge */
      </sw-mc-dp-blob-layout>
    </sw-mc-blob-layouts>
  </sw-mc-interface>

  <sw-mc-interface id="ADDRESS">
    <short-name>ADDRESS</short-name>
    <sw-mc-blob-layouts>
      <sw-mc-kp-blob-layout>
       long; /* Anfangs-Adresse */
      </sw-mc-kp-blob-layout>
    </sw-mc-blob-layouts>
  </sw-mc-interface>
</sw-mc-interface-spec>
```

### 5.4.2 Basetypes for mc interfaces

There are also basetypes for mc interfaces. These are referred in the blob layouts.

```
<sw-mc-base-types>
  <sw-mc-base-type id="LONG">
    <long-name>this is SW-MC-BASE-TYPE LONG</long-name>
    <short-name>long</short-name>
    <sw-base-type-size>32</sw-base-type-size>
    <sw-coded-type>signed</sw-coded-type>
    <sw-mem-alignment>32</sw-mem-alignment>
    <byte-order type="MOST-SIGNIFICANT-BYTE-LAST"></byte-order>
  </sw-mc-base-type>

  <sw-mc-base-type id="INT">
    <long-name>this is SW-MC-BASE-TYPE INT</long-name>
    <short-name>int</short-name>
    <sw-base-type-size>16</sw-base-type-size>
    <sw-coded-type>signed</sw-coded-type>
    <sw-mem-alignment>16</sw-mem-alignment>
    <byte-order type="MOST-SIGNIFICANT-BYTE-LAST"></byte-order>
  </sw-mc-base-type>

  <sw-mc-base-type id="CHAR">
    <long-name>this is SW-MC-BASE-TYPE CHAR</long-name>
    <short-name>char</short-name>
    <sw-base-type-size>8</sw-base-type-size>
    <sw-coded-type>signed</sw-coded-type>
    <sw-mem-alignment>8</sw-mem-alignment>
    <byte-order type="MOST-SIGNIFICANT-BYTE-LAST"></byte-order>
  </sw-mc-base-type>

  <sw-mc-base-type id="ASCII">
    <long-name>this is SW-MC-BASE-TYPE ASCII</long-name>
    <short-name>ascii</short-name>
    <sw-base-type-size>8</sw-base-type-size>
    <sw-coded-type>ascii</sw-coded-type>
    <sw-mem-alignment>8</sw-mem-alignment>
    <byte-order type="MOST-SIGNIFICANT-BYTE-LAST"></byte-order>
  </sw-mc-base-type>
</sw-mc-base-types>
```

### 5.4.3 Interface implementations

In the interface implementation, the TP-blob and the sources are instantiated.

```
<sw-mc-interface-impls>
  <sw-mc-interface-impl>
    <sw-mc-interface-ref id-ref="ETK">ETK</sw-mc-interface-ref>
    <sw-mc-tp-blob-conts>
      <sw-mc-blob-value>
        0x303FC0 0x1 0x27 0xC 2
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
    </sw-mc-blob-value>
  </sw-mc-tp-blob-conts>
```

### 5.4.4 Interface sources

Exactly as in *ASAM-MCD-2MC 1.x* the sources (indicating how often the MC-system should take samples on the hardware, not to be confused with how often the hardware updates the variables internally) has to be defined.

For each source, a short-name and a sample rate is defined. The **<sw-refresh-timing>** is using the CSE-codes defined by *[External Document: ASAP 1b specification / URL: / Relevant Position: ]*.

```
<sw-mc-interface-sources>
  <sw-mc-interface-source id=”IF-SYNCHRO”>
    <long-name>Combustion sychronous</long-name>
    <short-name>Synchro</short-name>
    <sw-refresh-timing>
      <sw-cse-code>103</sw-cse-code>
      <sw-cse-code-factor>1</sw-cse-code-factor>
    </sw-refresh-timing>

    <sw-mc-qp-blob-contents>
      <sw-mc-blob-value>
        <vf>0xa</vf>
      </sw-mc-blob-value>
      <sw-mc-blob-value>
        <vf>0x17F68</vf>
      </sw-mc-blob-value>
    </sw-mc-qp-blob-contents>
  </sw-mc-interface-source>

  <sw-mc-interface-source id=”IF-10MS”>
    <long-name>this is SW-MC-INTERFACE-SOURCE IF-10MS</long-name>
    <short-name>10ms</short-name>
    <sw-refresh-timing>
      <sw-cse-code>4</sw-cse-code>
      <sw-cse-code-factor>1</sw-cse-code-factor>
    </sw-refresh-timing>
    <sw-mc-qp-blob-contents>
      <sw-mc-blob-value>
        <vf>0x19</vf>
      </sw-mc-blob-value>
      <sw-mc-blob-value>
        <vf>0x17F7E</vf>
      </sw-mc-blob-value>
    </sw-mc-qp-blob-contents>
  </sw-mc-interface-source>

  <sw-mc-interface-source id=”IF-100MS”>
    <long-name>this is SW-MC-INTERFACE-SOURCE IF-100MS</long-name>
    <short-name>100ms</short-name>
    <sw-refresh-timing>
      <sw-cse-code>5</sw-cse-code>
      <sw-cse-code-factor>1</sw-cse-code-factor>
    </sw-refresh-timing>
    <sw-mc-qp-blob-contents>
      <sw-mc-blob-value>
        <vf>0x23</vf>
      </sw-mc-blob-value>
      <sw-mc-blob-value>
        <vf>0x17FB4</vf>
      </sw-mc-blob-value>
    </sw-mc-qp-blob-contents>
  </sw-mc-interface-source>
</sw-mc-interface-sources>
      </sw-mc-interface-impl>
    </sw-mc-interface-impls>
  </sw-mc-communication-spec>
</msrsw>
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# 6 The corresponding ASAM-MCD-2MC File

For reference purposes, we present the corresponding ASAM-MCD-2MC file here.

```
ASAP2_VERSION  1 31
/* Erstellt von DAMOS++ V1.15B-3 */
/begin PROJECT _70192 "Objektspeicher ME7.0"
  /begin HEADER ""
    VERSION    "_70192"
    PROJECT_NO _70192
  /end HEADER

  /begin MODULE DIM ""

  /begin A2ML
    taggedunion if_data {

/*********************************************************************/
/*                                                                   */
/*   ASAP2-Meta-Beschreibung für ETK                                 */
/*   - nur in ASAP V1.2 anwendbar -                                  */
/*                                                                   */
/*   Robert Bosch GmbH K3/EES4-Hünerfeld                             */
/*   Stand 0.2, 12.Aug.96                                            */
/*   Stand 0.3, 28.05.97 ASAP1B_ADDRESS                              */
/*                                                                   */
/*********************************************************************/

"ASAP1B_ETK" taggedstruct
{
  (block "SOURCE" struct
    {
      struct
      {
        char [100]; /* Source-Name */
        int;    /* Minimale Periodendauer gemessen in CSE */
        long;   /* Minimale Anzahl von Perioden */
      };
      taggedstruct
      {
        "QP_BLOB" struct   /* QP_BLOB fuer ASAP1b */
        {
          int;  /* Max. Groesse Displaytabelle */
          long; /* Adresse Displaytabelle */
        };
      };
    };
  ) *;    /* SOURCE kann mehrfach auftreten */

  "TP_BLOB" struct /* TP_BLOB fuer ASAP1b */
  {
    long; /* Trigger-Segment-Adresse */
    int;  /* Ausgabeformat der Displaytabelle 1=byte 2=word */
    int;  /* Trigger-Modus (spezielle Codierung) */
    int;  /* Typ der Displaytabelle (12, 11, 20) */
    int;  /* Byte-Reihenfolge 1 = high first, 2 = low first */
  };

  "DP_BLOB" struct /* DP_BLOB fuer ASAP1b */
  {
    long; /* Anfangs-Adresse */
    long; /* Laenge */
  };

  "KP_BLOB" struct /* KP_BLOB fuer ASAP1b */
  {
    long; /* Anfangs-Adresse */
    int;  /* Adress-Ort (0=intern, 1=extern) */
    int;  /* Anzahl der Bytes */
  };
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
};

"ASAP1B_ADDRESS" taggedstruct
{
  "KP_BLOB" struct /* KP_BLOB fuer ASAP1b */
  {
    long; /* Anfangs-Adresse */
  };

};

/*  CMS REPLACEMENT HISTORY, Element ETK.AML */
/*  *5    27-JUN-1997 11:30:01 K3EES4_HD "'block' bei KB_BLOB entfernt" */
/*  *4    28-MAY-1997 15:38:29 K3EES4_HD "leeren PA_BLOB entfernt" */
/*  *3    28-MAY-1997 10:41:57 K3EES4_BA "Erweiterungen zu ASAP1B_ADDRESS" */
/*  *2    12-AUG-1996 13:25:27 K3EES4_HD "Neustrukturierung aufgrund Volvo-Besuch 1.8.96" */
/*  *1    16-JUL-1996 14:54:49 K3EES4_HD "ASAP-Meta-Language für ETK" */
/*  CMS REPLACEMENT HISTORY, Element ETK.AML */

    };
  /end A2ML

  /begin MOD_PAR "Objektspeicher ME7.0"
    VERSION "1.65"
    ADDR_EPK 0xFFFF
    EPK ""
    CUSTOMER_NO "MSR SW Working group"
    USER "Herr Smith"
    PHONE_NO ""
    CPU_TYPE "S80166"
    ECU ""
    /begin MEMORY_SEGMENT
      mem-seg-1
      ""
      CODE
      ROM
      INTERN
      0x0 0xC000 -1 -1 -1 -1 -1
      /begin
        IF_DATA ASAP1B_ETK DP_BLOB 0x0 0xC000
      /end IF_DATA
    /end MEMORY_SEGMENT

    /begin MEMORY_SEGMENT
      mem-seg-2
      ""
      DATA
      RAM
      INTERN
      0xC000 0x2000 -1 -1 -1 -1 -1
      /begin IF_DATA
        ASAP1B_ETK DP_BLOB 0xC000 0x2000
      /end IF_DATA
    /end MEMORY_SEGMENT

    SYSTEM_CONSTANT "EPK_MAX_LEN" "80"
    SYSTEM_CONSTANT "SWOFFDL_REQUEST_MASK" "16383"
    SYSTEM_CONSTANT "SY_2SG" "0"
  /end MOD_PAR

  /begin MOD_COMMON ""
    BYTE_ORDER MSB_LAST
    S_REC_LAYOUT DAMOS
  /end MOD_COMMON

  /* — AgI-Objekt für ETK (ASAP V1.2), Stand 1.0 27.6.97 — */
  /begin IF_DATA ASAP1B_ETK
      TP_BLOB  0x303FC0  0x1  0x27  0xC  2

      /begin SOURCE "Synchro" 103 1
        QP_BLOB  0xA  0x17F68
      /end SOURCE

      /begin SOURCE "10ms" 4 1
        QP_BLOB  0x19  0x17F7E
      /end SOURCE
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
    /begin SOURCE ”100ms” 5 1
      QP_BLOB  0x23  0x17FB4
    /end SOURCE
/end IF_DATA


/begin CHARACTERISTIC
    ABGMSIGH
    ”threshold for exh. temp. for wiring-interruption with Ri-diagnosis downstr. cat”
    VALUE
    0x1A0A8
    KwUb
    1275.
    temp_ub_q5_o50
    -50.00
    1225.

    FORMAT ”%5.0”

    /begin IF_DATA
      ASAP1B_ETK  DP_BLOB 0x1A0A8  0x1
    /end IF_DATA

/end CHARACTERISTIC

/begin RECORD_LAYOUT KwUb

    FNC_VALUES        1 UBYTE COLUMN_DIR DIRECT

/end RECORD_LAYOUT

/begin COMPU_METHOD

    temp_ub_q5_o50
    ””
    RAT_FUNC
    ”%6.2”
    ”Grad C”

    COEFFS 0 1 50 0 0 5

/end COMPU_METHOD



/begin CHARACTERISTIC

    ANALUN
    ”number of combustions for deactivation after detected misfire”
    CURVE
    0x1125C
    GklWUb
    255.0
    dez
    0.00
    255.0

    FORMAT ”%5.1”

    /begin IF_DATA
      ASAP1B_ETK  DP_BLOB 0x1125C  0x8
    /end IF_DATA

    /begin AXIS_DESCR

        COM_AXIS
        nmot
        nmot_ub_q40
        8
        0.00
        10200.

        AXIS_PTS_REF SNM08DMUB

    /end AXIS_DESCR
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
/end CHARACTERISTIC

/begin RECORD_LAYOUT GklWUb

    FNC_VALUES        1 UBYTE COLUMN_DIR DIRECT

/end RECORD_LAYOUT

/begin COMPU_METHOD

    dez
    "1 : 1 conversion (dez/hex)"
    RAT_FUNC
    "%6.2"
    ""

    COEFFS 0 1 0 0 0 1

/end COMPU_METHOD

/begin COMPU_METHOD

    nmot_ub_q40
    ""
    RAT_FUNC
    "%6.1"
    "Upm"

    COEFFS 0 1 0 0 0 40

/end COMPU_METHOD

/begin AXIS_PTS

    SNM08DMUB
    ""
    0xD91A
    nmot
    SstAUbSstUb
    10200.
    nmot_ub_q40
    8
    0.00
    10200.

    DEPOSIT ABSOLUTE

    /begin IF_DATA
      ASAP1B_ETK  DP_BLOB 0xD91A  0x9
    /end IF_DATA

/end AXIS_PTS

/begin RECORD_LAYOUT SstAUbSstUb

    NO_AXIS_PTS_X     1 UBYTE
    AXIS_PTS_X        2 UBYTE INDEX_INCR DIRECT

/end RECORD_LAYOUT




/begin CHARACTERISTIC

    FKKVS
    "factor to correct fuel delivery system"
    MAP
    0x1FA40
    KfAxUbAyUbSstxUbSstyUwWUw
    1.999969
    fak_uw_b2
    0.00
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
    1.999969

    FORMAT "%8.6"

    /begin IF_DATA
      ASAP1B_ETK  DP_BLOB 0x1FA40  0x232
    /end IF_DATA

    /begin AXIS_DESCR

        STD_AXIS
        nmot
        nmot_ub_q40
        16
        0.00
        10200.

        DEPOSIT ABSOLUTE

    /end AXIS_DESCR

    /begin AXIS_DESCR

        STD_AXIS
        tevfa_w_kge
        tinjection_sy
        16
        0.00
        131.0700

        DEPOSIT ABSOLUTE

    /end AXIS_DESCR

/end CHARACTERISTIC

/begin RECORD_LAYOUT KfAxUbAyUbSstxUbSstyUwWUw

    NO_AXIS_PTS_X     1 UBYTE
    NO_AXIS_PTS_Y     2 UBYTE
    AXIS_PTS_X        3 UBYTE INDEX_INCR DIRECT
    AXIS_PTS_Y        4 UWORD INDEX_INCR DIRECT
    FNC_VALUES        5 UWORD ROW_DIR DIRECT

/end RECORD_LAYOUT

/begin COMPU_METHOD

    fak_uw_b2
    ""
    RAT_FUNC
    "%7.6"
    "-"

    COEFFS 0 65536 0 0 0 2

/end COMPU_METHOD

/begin COMPU_METHOD

    tinjection_sy
    ""
    RAT_FUNC
    "%6.4"
    "ms"

    COEFFS 0 500 0 0 0 1

/end COMPU_METHOD



/begin MEASUREMENT
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
    nmot
    "engine speed"
    UBYTE
    nmot_ub_q40
    1
    100
    0
    10200

    FORMAT "%5.0"

    /begin IF_DATA
      ASAP1B_ADDRESS  KP_BLOB 0xF86C
    /end IF_DATA
    /begin IF_DATA
      ASAP1B_ETK  KP_BLOB 0xF86C 0x0 0x1
    /end IF_DATA
/end MEASUREMENT

/begin MEASUREMENT

    B_2ph
    "Condition for the second phase edge"
    UWORD
    B_TRUE
    1
    100
    0
    1

    BIT_MASK 0x800
    FORMAT "%13.11"

    /begin IF_DATA
      ASAP1B_ADDRESS  KP_BLOB 0xFD0C
    /end IF_DATA
    /begin IF_DATA
      ASAP1B_ETK  KP_BLOB 0xFD0C 0x0 0x2
    /end IF_DATA
/end MEASUREMENT

/begin COMPU_METHOD
    B_TRUE
    ""
    TAB_VERB
    "%6.3"
    ""
    COMPU_TAB_REF B_TRUE

/end COMPU_METHOD

/begin COMPU_VTAB

    B_TRUE
    ""
    TAB_VERB
    2
    0 "--"
    1 "TRUE"

/end COMPU_VTAB

/begin MEASUREMENT

    tevfa_w_kge
    ""
    UWORD
    tinjection_sy
    1
    100
    0
    131.07

    FORMAT "%8.4"

    /begin IF_DATA
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

```
        ASAP1B_ADDRESS  KP_BLOB 0x3012A2
      /end IF_DATA
      /begin IF_DATA
        ASAP1B_ETK  KP_BLOB 0x3012A2 0x1 0x2
      /end IF_DATA

  /end MEASUREMENT



  /end MODULE
/end PROJECT
```

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# 7 The Road ahead

## 7.1 ASAM-MCD-2MC: An important chapter in the full story

This document describes merely the ASAM-MC usecase. But the DTD can be applied to any step in the entire engineering Process. Since not all Information is required or available at any step, the population of the defined structure will vary from step to step (see Figure 17 Using the DTD in multiple process steps p. 62).

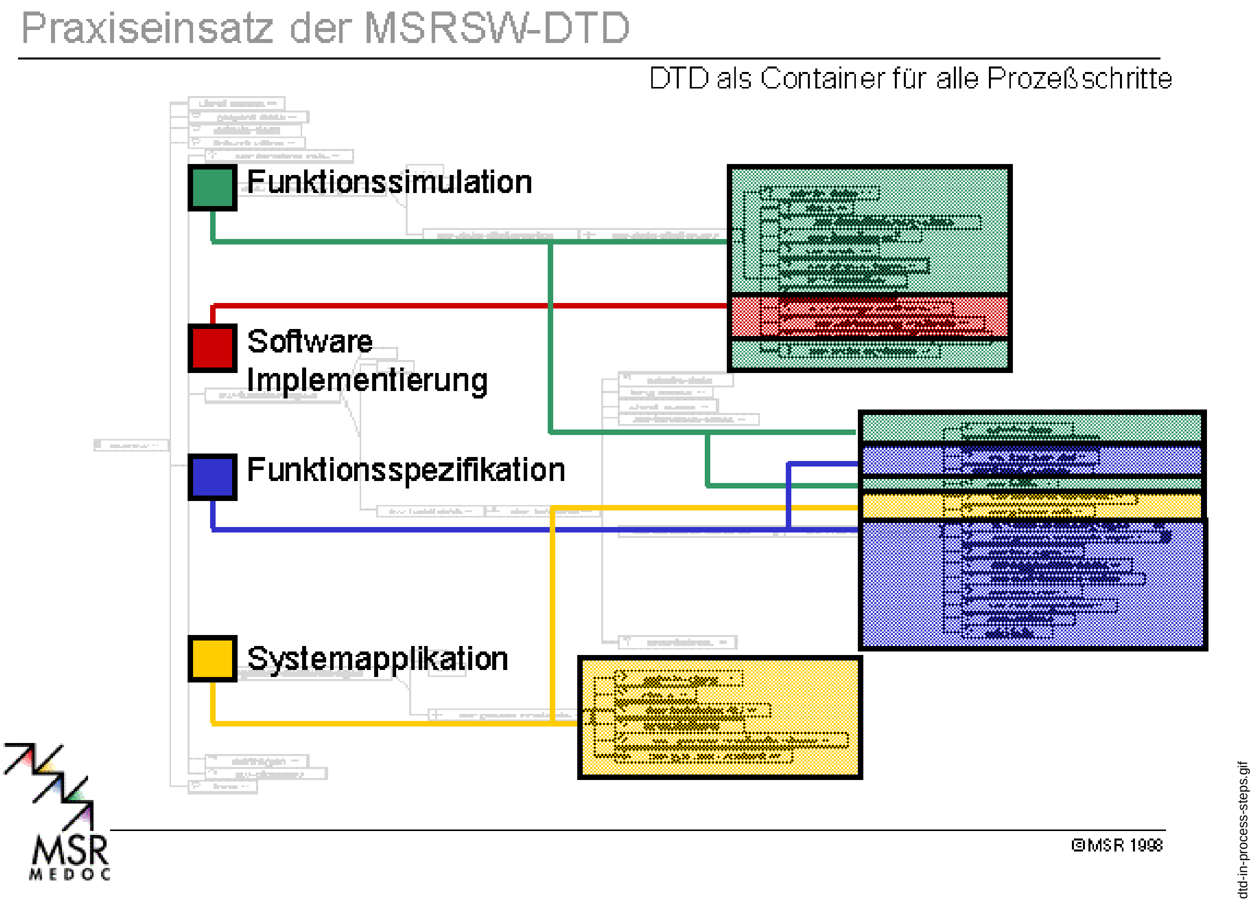



**Figure 17: Using the DTD in multiple process steps**

For this reason, the actual DTD contains more elements than being described in this document. It even contains more features used by the ASAM-MC usecase which are not described within this introductory document.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

### 7.1.1 The Document Content Information

As the DTD may be used in many different use cases, there must be additional means for defining the use cases independent from the DTD. In addition to this it is necessary to define the files more precisely as it is possible in an SGML DTD such as datatypes specific structures etc.

This definition can be done in the *DCI*. The *DCI* takes approaches from the actual schema and datatype discussion in the W3C In particular, the DCI provides the following features:

- Predefined datatypes - these datatypes can be applied to particular elements. These data types can be:
    - basic data types, e.g. integer, long, byte, etc.
    - lexical data types defined by regular expressions.
    - user defined data-types can be checked by giving specific scripts
    - lists of predefined values
- List of possible values - these lists can be used to check the contents of particular elements.
- There are means to express the length of the particular element content.
- Define specific substructures controlled by ”class-elements” e.g. **<sw-param-class>**. This allows to specify precisely, what kind of information must be there for e.g. a map.
- All definitions can be specified by populating a predefined structure as well as by applying scripts to it. The scripts may be specified simultaneously in multiple scripting languages such as *vbscript*, *mmx*, *javascript*.
- All definitions can be checked by a generic dci checker which interprets a DCI file[11].

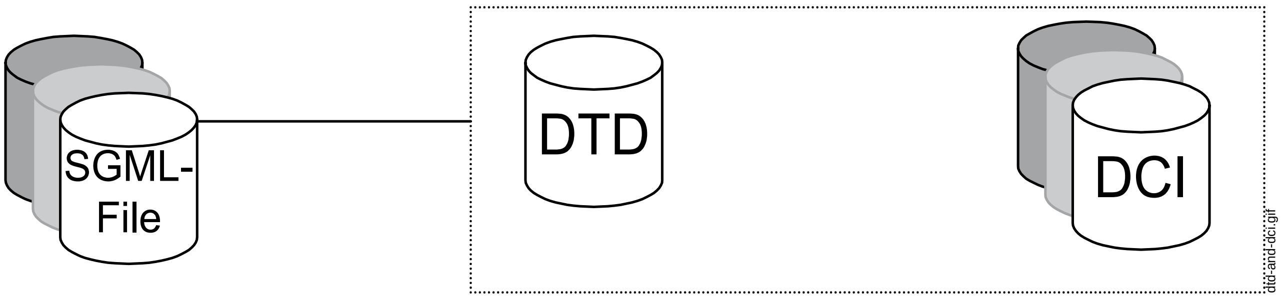



**Figure 18: SGML-Instance, DTD and DCI**

In order to achieve robust processes, the whole picture consists of three files:

- The DTD which defines the basic structure. The DTD tells, where the information can be found if it is there. The DTD stays the same for all use cases.
- The DCI which defines the amount of information for a particular usecase. Usually there are more *DCI*s than there are DTDs. Without a DCI, this information must be provided as prose in the accompanying documentation of the standards.

[11] this checker is not yet implemented.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

- The data file itself which contains the actual information. This file can be checked to match the DTD and the DCI. If this check is successful, it can be used without further checking.

## 7.2 Classes and functions which are features

One additional feature (compared to *ASAM-MCD-2MC 1.x* and *MSRSW V1.1.0*) is the approach to introduce classes and multiple instantiation.

The concept of classes is introduced as an enhancement to the concepts of parameters and variables provided up to now[12]. Therefore we have:

- Features[13]. of a software system. Such a feature exists exactly once within a software system. It can be setup by other features thus allowing to define a hierarchical software system architecture.
- calibration parameters (**<sw-params>**) and variables (**<sw-variables>**) which can be defined in a global namespace within the global data dictionary, as well as in a local namespace within a data dictionary which is defined locally within a particular feature.

  Local and global definition solely covers namespaces. The visibility etc. is handled with in the feature accessing the variables.

  New in this context is the option to define variables and parameters in a local or global data dictionary.
- As a new concept, a software feature can be defined by instantiating classes. A class (**<sw-class>**) is unique within a software system, but can be instantiated more than once either within a feature or within the global data dictionary.

  This multiple instantiation makes causes the fact that the ”structure information” no longer matches one to one to the instance tree.

In Figure 19 Classes and multiple instantiation p. 66 the concepts are visualized:

- The global data dictionary defines the variable ”A” and the parameter ”B” and the object ”GO” of class ”C”.
- in **<sw-feature-spec>** there is a feature ”F” whose local data dictionary defines an object ”FO” of class ”C”.
- The class ”C” is defined within **<sw-component-spec>**, and comprises of a variable ”CA” and a parameter ”CB”. This means, that every instance of this class will instantiate its own ”CA” and ”CB”.
- In **<sw-instance-spec>** one can see all instances:

  A This is the global variable **”A”**, defined in the global data dictionary

  B This is the global calibration parameter **”B”**, defined in the global dictionary

[12] Although it would be possible to express everything as classes, the workgroup decided to keep the existing concepts for sake of easy understanding.

[13] formerly called function

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

GO.CA This is the global variable **"CA"** which is part of the global object **"GO"** which is an instance of class **"C"**

GO.CB This is the global calibration parameter **"CB"** which is part of the global object **"GO"** which is an instance of class **"C"**

F.FO.CA This is the local variable **"CA"** which is part of the local object **"FO"** which is an instance of class **"C"** within feature **"F"**

F.FO.CB This is the local calibration **"CB"** which is part of the local object **"FO"** which is an instance of class **"C"** within feature **"F"**

For sake of clarity, the names in this example are "flattened". In the *MSRSW.DTD*, the instances are denoted as a real tree. This allows to specify information at any level of the instance tree.

The example shows that there are local namespaces. This allows to use any name for local items within a class or a feature regardless of possible naming conflicts with global items. This is because classes and features establish their own namespace.

Classes not only may comprise of variables or parameters but also of other class instances. This is not shown in the example.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

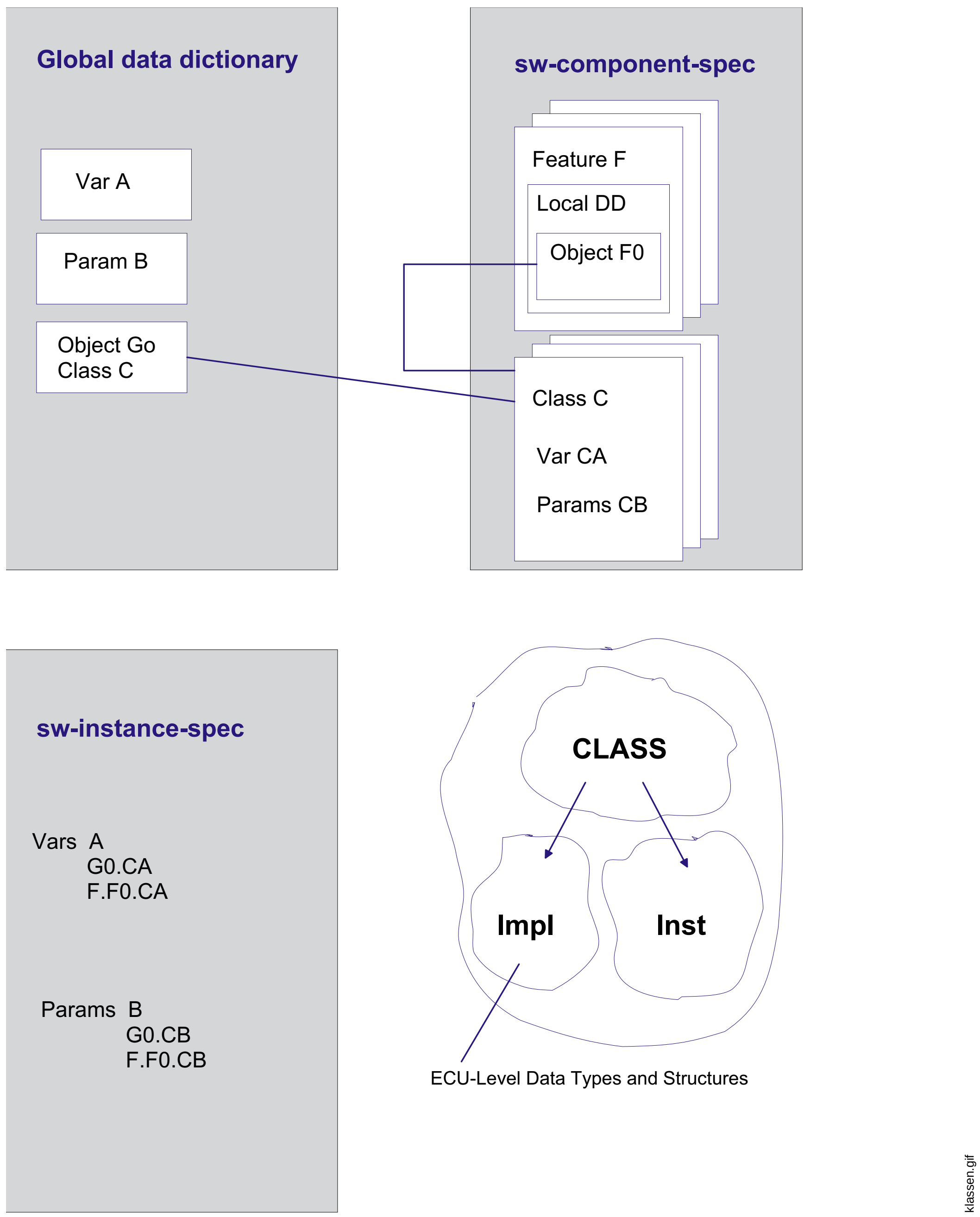



**Figure 19: Classes and multiple instantiation**

## 7.3 More details on data models

This chapter shows in more detail the most important data models of the DTD. [14]

***Hint:***
The figures in particular still reflect to prerelease 4. But the overall architecture has not changed.

### 7.3.1 Data model for the data dictionary

This data model shows the data dictionary. A short description of the associations is given below. Please refer also to the previous chapters Topic 3 Basic Data model regarding *ASAM-MCD-2MC* usecase p. 23 and Topic 5 Annotated Sample file p. 37.

[14] Note that these illustrations are not fully checked and probably not fully complete. They are presented here to give a first impression and to gain feedback from the audience if this presentation is helpful.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

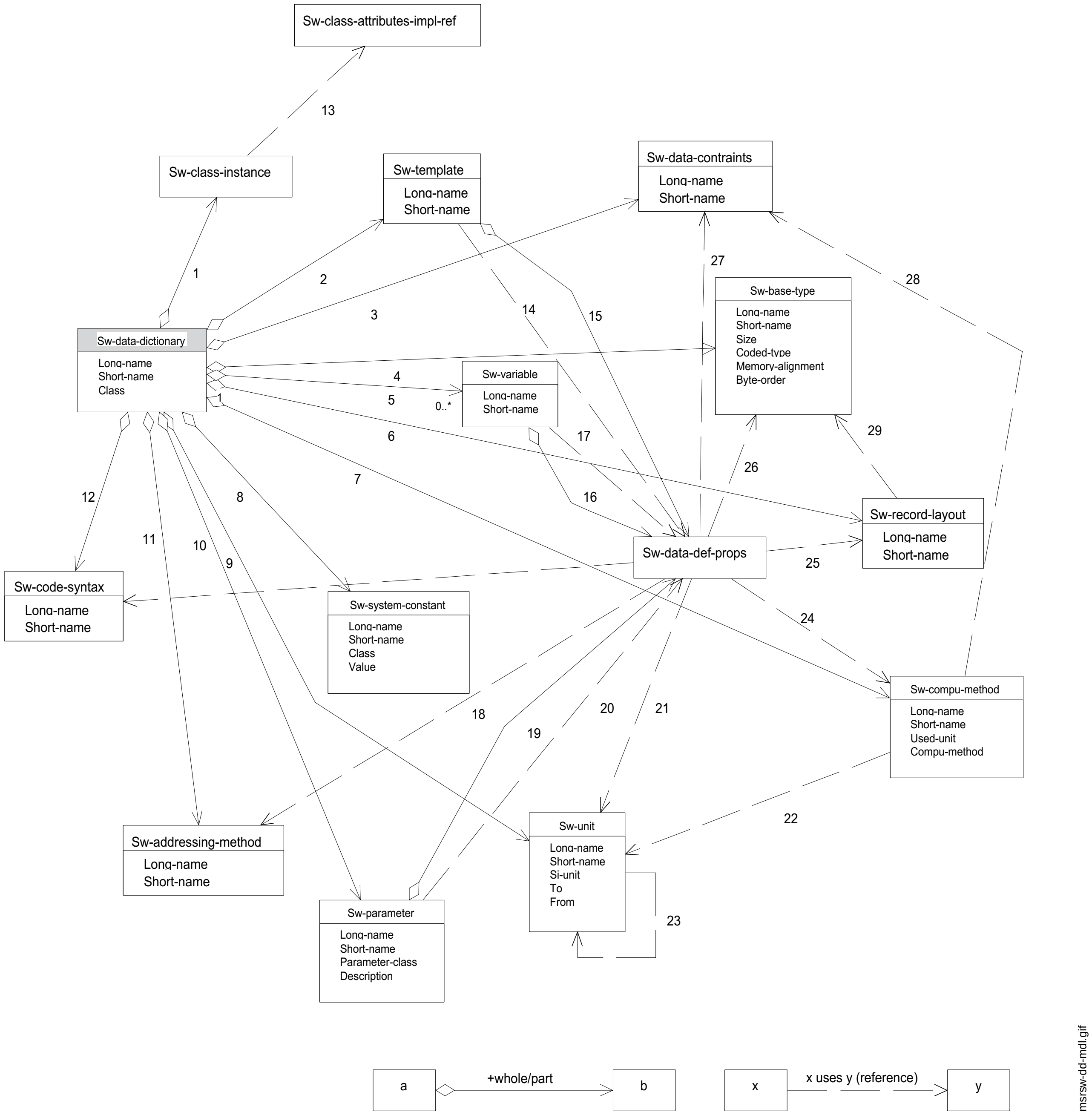



**Figure 20: The data model for the data dictionary**

The following short description of the data model refers the callouts in the Figure 20 The data model for the data dictionary p. 68. In the models we have two types of associations[15]:

---

[15] compare to UML

a "whole-part"-association and a "uses"-association which is established by references (usually done using *-ref elements in the dtd).

(1): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-class-instance>**. This is a particular global or local instance of a **<sw-class>**. The contents of this element superseed the contents of the referenced **<sw-class>** thus defining the particular implementation.

(2): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-template>**. These Templates are used to predefine some properties of variables or parameters. The templates can be used to simplify the typing task. They have no influence on the data model in the ecu.

(3): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-sw-data-constr>**. These constraints may be applied to data object such as computation methods, variables and parameters.

(4): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-base-type>**. These base-types are used by variables, parameters, and record layouts (Topic 3.1.1 Base types p. 27).

(5): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-variable>** which represent variables in the ecu (Topic 3.1.4 Variables and parameters p. 29).

(6): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-record-layout>** which are used by variables and parameters (Topic 3.1.5 Record Layouts p. 30 resp. Topic 5.3.1.6 Record layouts p. 44).

(7): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-compu-method>** which are used by variables and parameters (Topic 3.1.3 Computation methods p. 28 resp. Topic 5.3.1.5 Computation methods p. 42).

(8): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-system-constant>**. These system constants can be used to define a configurable ASAM-MCD-2MC-file. Therefore the sytem constants are referenced e.g. within an arraysize.

(9): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-unit>** which are used by computation methods etc. See also Topic 3.1.2 Physical units p. 27 resp. Topic 5.3.1.1 Measurement units p. 38.

(10): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-calprm>** which represent calibration parameters in the ecu (Topic 3.1.4 Variables and parameters p. 29 resp. Topic 5.3.1.3 Calibration parameters p. 40).

(11): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-addressing-method>**.

(12): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-code-syntax>**.

(13): A **<sw-class-instance>** uses **<sw-class-attributes-impl>** defined within **<sw-class>** by reference **<sw-class-attributes-impl-ref>**. This selects a particular one of the possible implementations of the class.

(14): A template, variable or parameter can be defined based on templates. Therefore **<sw-data-def-props>** references a template [16]

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

(15): A **<sw-template>** contains optionally a **<sw-data-def-props>**. By this, the properties leveraged to the references of the templates are specified.

(16): A **<sw-variable>** contains optionally a **<sw-data-def-props>**. By this, the properties of the variable are specified.

(17): This reference does not really exist[17].

(18): In a **<sw-data-def-props>** a **<sw-addressing-method>** can be used by inserting a element **<sw-addressing-method-ref>** which contains the short-name of the referenced **<sw-addressing-method>**. This denotes, by which method the object in question (represented by the parent of **<sw-data-def-props>**) can be adressed in the ecu.

(19): A **<sw-calprm>** contains optionally a **<sw-data-def-props>**. By this, the properties of the variable are specified.

(20): This reference does not really exist[18].

(21): In **<sw-data-sef-props>** a **<sw-unit>** can be referenced/used by using the element **<sw-unit-ref>**. This denotes, which measurement unit applies to the object in question (represented by the parent of **<sw-data-def-props>**). By this it is possible to define data objects on a pure physical level, in particular without a computation method.

(22): In **<sw-compu-method>** a **<sw-unit>** can be referenced/used by using the element **<sw-unit-ref>**. This denotes, which measurement unit applies to computation method.

(23): Reference of a **<sw-unit>** to a base unit ( which is a **<sw-unit>** containing a **<si-unit>** ). This approach allows handle and to convert values and their measurement units formally.

(24): In **<sw-data-def-props>** a **<sw-compu-method>** can be applied to the object in question (denoted by the container of **<sw-data-def-props>**) by using the element **<sw-compu-method-ref>**.

(25): In **<sw-data-def-props>** a **<sw-record-layout>** can be referenced/used by using the element **<sw-record-layout-ref>**. This denotes how the object in question (denoted by the container of **<sw-data-def-props>**) is written by a code generator[19].

(26): In **<sw-data-def-props>** a **<sw-base-type>** can be referenced/used by using the element **<sw-base-type-ref>**. This denotes of which base typ the object in question (denoted by the container of **<sw-data-def-props>**) is in the ecu.

(27): In **<sw-data-def-props>** a **<sw-data-contraint>** can be referenced/used by using the element **<sw-data-contraints-ref>**. These constraints apply for example in the

[16] note that the arrow is of wrong direction in the graphic.

[17] this is a bug in the figure.

[18] this is a bug in the figure.

[19] this is a typical application of an early phase in the development process.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

calibration phase to the object in question (denoted by the container of **<sw-data-def-props>**). These constraints can also be used to calibrate a graphical display of a curve etc.

(28): In **<sw-compu-method>** a **<sw-data-constr>** can be referenced/used by using the element **<sw-data-contraints-ref>**. These constraints represent the set of input values for which the computation method can be applied. These constraints are used for example to prevent from divisions by zero or to cut a monotoneous segment from a higher order conversion formula.

(29): In **<sw-record-layout>** a **<sw-base-type>** can be referenced/used by using the element **<sw-base-type-ref>**. This denotes how the object in question (denoted by the container of **<sw-data-def-props>**) is stored in the memory of the ecu.

### 7.3.2 Data model for features

This data model shows software components (primarily featuers). A short description of the associations is given below.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

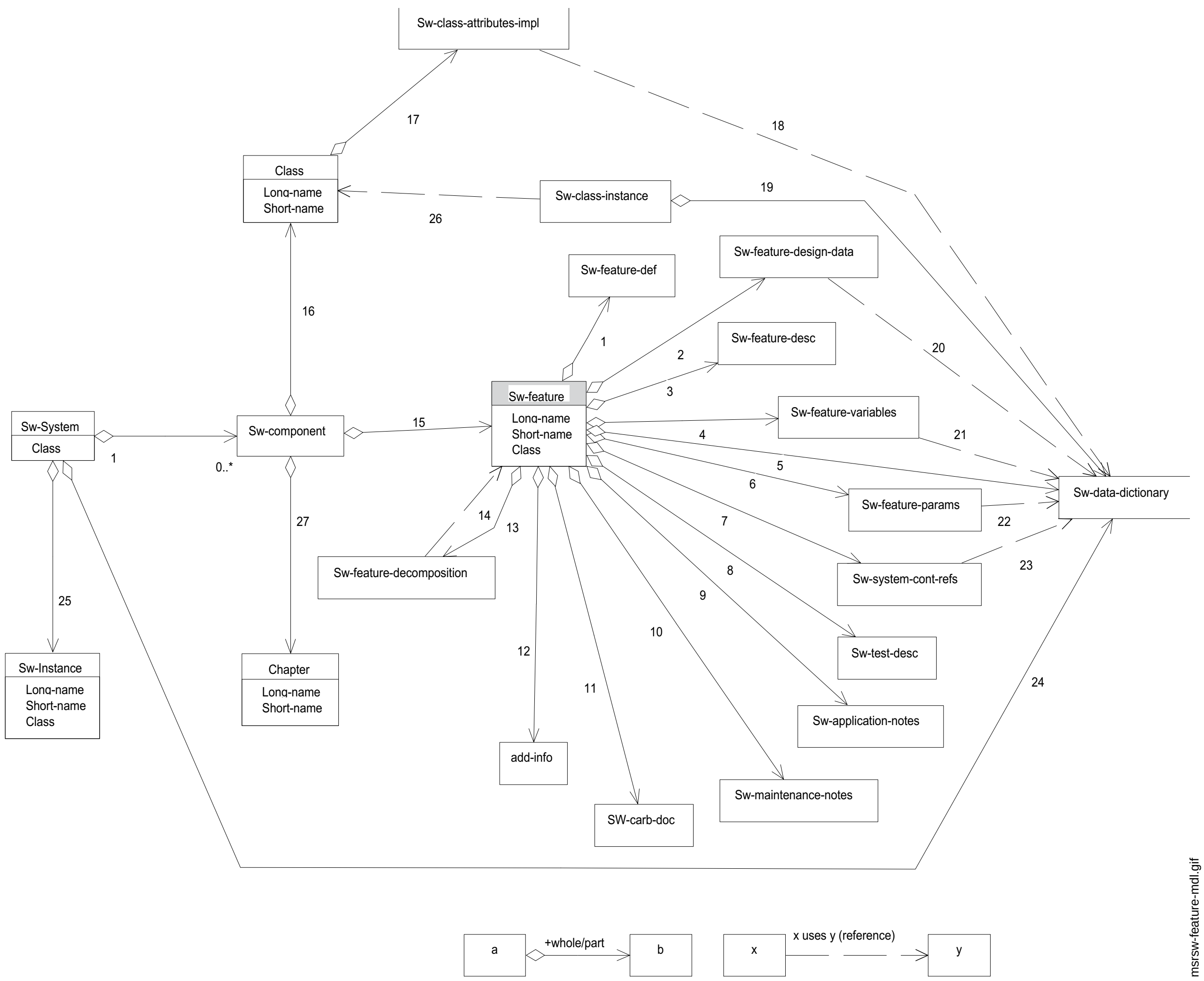



**Figure 21: Data model for features**

(1): A **<sw-feature>** contains optionally 1 **<sw-feature-def>.** This is bound to contain formal definitions of the feature. Usually this is populates using *SYSTEM design tool* s like *ASCET/SD* or *MATRIX-X*

(2): A **<sw-feature>** contains optionally 1 **<sw-feature-design-data>**. This branch is bound for the early design phase, where it is not yet decided how the module architecture will be setup.

(3): A **<sw-feature>** contains optionally 1 **<sw-feature-desc>**. This is bound for exhaustive description of the feature and is usually filled using an *SGML-editor*.

(4): A **<sw-feature>** contains optionally 1 **<sw-feature-variables>** containing references to variables. This represents the relationship of the feature to the referenced variables. In particular, this branch allows to define which variables are instantiated by the particular feature as well as how the variable is accessed (read, write etc). As the

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

DTD is used up to now these references to variables are the point of instantiation of the variables.

(5): A **<sw-feature>** contains optionally 1 **<sw-data-dictionary>**. This dictionary establis a namespace local to the feature. Therefore it allows to define feature specific data objects[20].

(6): A **<sw-feature>** contains optionally 1 **<sw-feature-params>** containing references to calibration parameters. This represents the relationship of the feature to the referenced parameters. In particular, this branch allows to define which variables are instantiated by the particular feature. As the DTD is used up to now these references to parameters are the point of instantiation of the parameters.

(7): A **<sw-feature>** contains optionally 1 **<sw-system-const-refs>**. It is a container for **<sw-system-const-ref>**. This allows features to introduce (instantiate) or to refer to system constants.

(8): A **<sw-feature>** contains optionally 1 **<sw-test-desc>**. This element can be used to describe test procedures for the particular feature.

(9): A **<sw-feature>** contains optionally 1 **<sw-application-notes>**. This can be used to describe specific calibration procedures for the particular feature.

(10): A **<sw-feature>** contains optionally 1 **<sw-maintenance-notes>**. This can be used to describe specific maintenance procedures for the particular feature.

(11): A **<sw-feature>** contains optionally 1 **<sw-carb-doc>**. This can be used to provide specific documentation according to *CARB*.

(12): A **<sw-feature>** contains optionally 1 **<add-info>** allowing to provide additional information for which no speicific element exists in the DTD.

(13): A **<sw-feature>** contains optionally 1 **<sw-feature-decompositon>**. This is a set of links to other features establishing the actual feature. As an example Lambda-Control consists of Lambda-measurement and injection-corretion. This allows a hierarchical system decomposition[21].

(14): This is a link to another feature which establishes the fact that the referenced feature is part of the decomposition of the actual feature.

(15): A software system is setup of components. Therefore **<sw-components>** may contain up to n **<sw-feature>**s.

(16): The features of a system may be setup using classes. Therefore A **<sw-components>** may contains up to n **<sw-class>**.

(17): A class may appear in various implementations within a systems. Therefore **<sw-class>** contains 1 or n **<sw-class-attributes-impl>** each of representing a particular Implementation of a class.

[20] note that these variables may still be publicly visible. They can be referred to by the msr natural adressing through prefixing with the feature name.

[21] note that in further revisions more information will be provided here.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

(18): The **<sw-class-attributes-impl>** are referenced by the instantiations (**<sw-class-instance>**) of the class within **<sw-data-dictionary>** [22].

(19): A **<sw-data-dictionary>** contains optionally 1 to n **<sw-class-instance>**s. This establishes instantiations of the class.

(20): In **<sw-feature-design-data>** variables, parameters of a **<sw-data-dictionary>** are referenced. This is bound for the very early phases in the software engineering process where the architecture is not fully decided and therefore the information required in **<sw-feature-variables>** etc. is not available.

(21): Links to particular variables in the data dictionary used by the **<sw-feature>**.

(22): Links to particular calibration parameters in the data dictionary used by the **<sw-feature>**.

(23): Links to particular system-constants in the data dictionary used by the **<sw-feature>** .

(24): A **<sw-system>** contains 0 .. n **<sw-data-dictionary>** representing the global data dictionary. Note that the purpose of the data dictionary can be specified in **<sw-data-dictionary-class>** [23].

(25): A **<sw-system>** contains 0 .. n **<sw-instance>** which represent instance specific information (Topic 3.3 The instance-tree p. 31).

(26): This is a particular instantiation of a sw-class. The particular implementation for the instance can be choosen from the implementations provided by the class (**<sw-class-attributes-impl-ref>**).

[22] todo: note that the direction of the reference in the figure is wrong.

[23] more details are defined in the DCI

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

### 7.3.3 Data model for classes

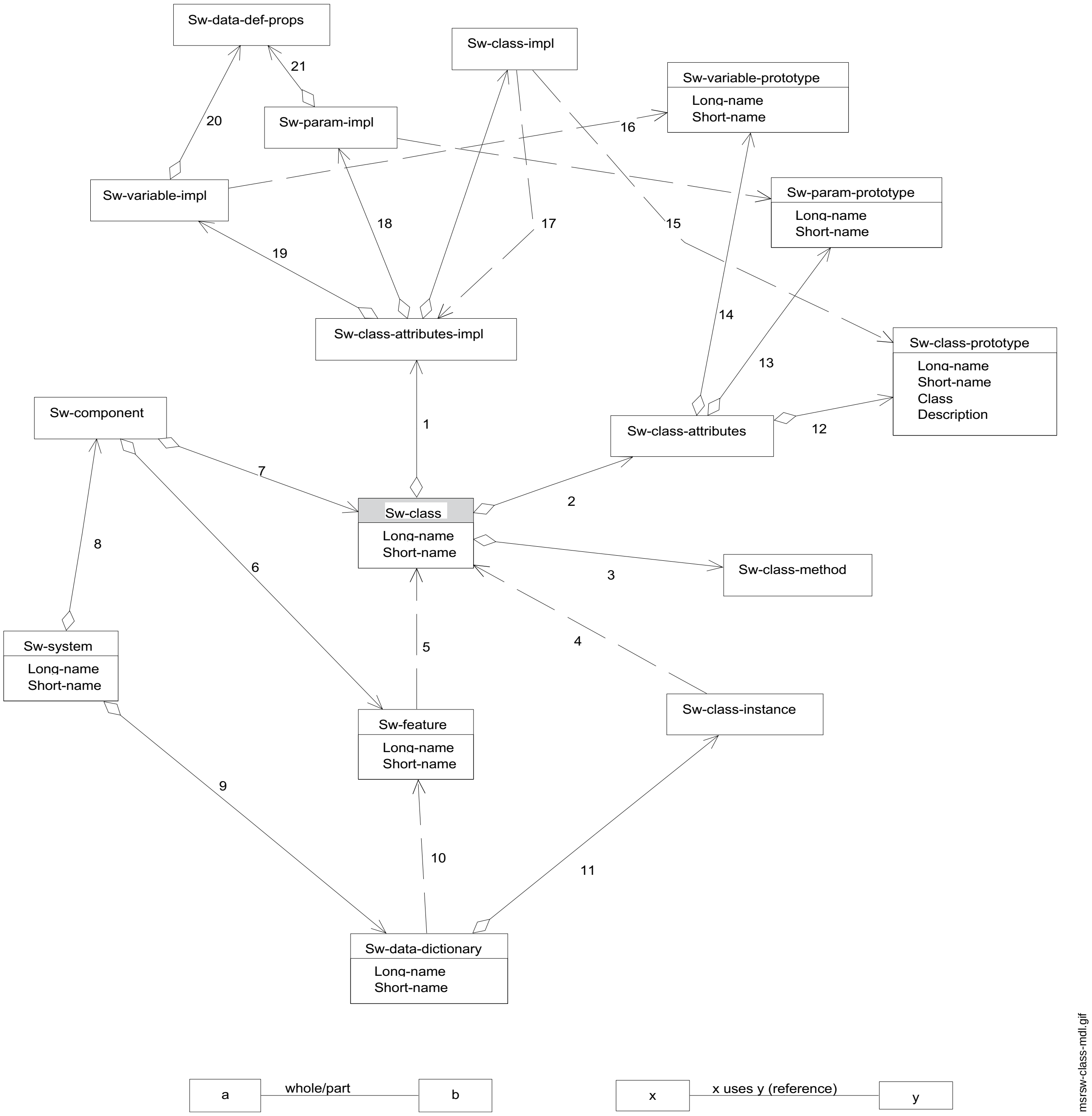



**Figure 22: Data model for classes**

(1): A class may provide multiple implementations. In *MSRSW.DTD* this primarily covers the implementation of variables and parameters which come from the instantiation of this class. A **<sw-class>** therefore contains optionally 1 or n **<sw-class-attributes-impl>**, each of which representing a particular implementation of a class.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

(2): A **<sw-class>** contains optionally 0 or n **<sw-class-attributes>**. This is merely the list of variables, parameters and objects which will be established by the instantiation of the class. This is one of the reasons, that we need an **<sw-instance-tree>**

(3): A **<sw-class>** contains optionally 0 or n **<sw-class-method>**, each of which representing a particular method of a class. The methods are described informally. This branch is bound to future extensions.

(4): Classes are instantiated within the data dictionaries, either the global one or within a feature. Therefore a **<sw-class-instance>** in a **<sw-data-dictionary>** has a reference to a **<sw-class>**. This represents a particular instance of a **<sw-class>**. The contents of this element superseed the contents of the referenced **<sw-class>** thus defining the particular implementation.

(5): Todo: this reference is a bug in the figure.

(6): A software system is setup of components. Therefore **<sw-components>** may contain up to n **<sw-feature>**s.

(7): A software system is setup of components. Therefore **<sw-components>** may contain up to n **<sw-class>**s.

(8): A **<sw-system>** contains optionally 1 **<sw-components>**. This is the part, where any features/functions/classes of a software system are specified. It is also possible to insert **<chapter>**s in **<sw-components>** allowing to establish intermediate documentation chapters.

(9): A **<sw-system>** may contain multiple **<sw-data-dictionary>**. These represent the global data dictionary. The particular purpose of particular data dictionaries is denoted in **<sw-data-dictionary-class>**.

(10): A **<sw-data-dictionary>** may optionally be linked to a **<sw-feature>**. This is for historical reasons and allows to establish an association between features and data dictionaries for administrative purposes.

(11): A **<sw-data-dictionary>** contains optionally 1 or n **<sw-class-instance>**. This establishes instances of classes in the context of the parent data dictionary (either global or local to a feature).

(12): **<sw-class-attributes>** contains optionally 1 or n **<sw-class-prototype>**. This denotes attributes within the classes which themselves are instances of classes. The particular implementation is selected during the instantiation of the class (**<sw-class-instance>**).

(13): **<sw-class-attributes>** contains optionally 1 or n **<sw-param-prototype>**. This denotes attributes within the classes which are calibration parameters. The particular implementation is selected during the instantiation of the class (**<sw-class-instance>** ).

(14): **<sw-class-attributes>** contains optionally 1 or n **<sw-variable-prototype>**. This denotes attributes within the classes which are variables. The particular implementation is selected during the instantiation of the class (**<sw-class-instance>**).

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

(15): **<sw-class-impl>** refers to a particular attribute prototype of the class (**<sw-class-prototype-ref>**) in order to specify the implementation details for this prototype within a particular implementation provided by the class.

(16): A **<sw-variable-impl>** contains references to a of the class (**<sw-variable-prototype-ref>**) in order to specify the implementation details for the prototype within a particular implementation provided by the class. The same applies to **<sw-param-impl>** (todo: which is has no callout in the graphic).

(17): When a class-prototype is instantiated, the implementation for the particular prototype (denoted by reference 15) must be selected. Therefore **<sw-class-impl>** contains optionally ontains a reference to **<sw-class-attributes-impl>**. The two references (15 and 17) must end in the same **<sw-class>** (via 12, 2, respectively 1).

(18): The **<sw-class-attributes-impl>** contains optionally 1 or n **<sw-variable-impl>**. These denote the implementation of the attributes which are variables.

(19): The **<sw-class-attributes-impl>** contains optionally 1 or n **<sw-variable-impl>**. These denote the implementation of the attributes which are calibration parameters.

(20): A **<sw-variable-impl>** contains optionally a **<sw-data-def-props>** which contain all the implementation details of the variable.

(21): A **<sw-param-impl>** contains optionally a **<sw-data-def-props>** which contains all the implementation details of the calibration parameter.

## 7.4 The documentation aspect

The DTD provides further means to integrate documentation, especially in the elements within **<sw-feature>**. This can be done as prose because these elements follow the model of **<ncoi>** (which stands for non content oriented information[24]). Basically **<ncoi>** provides the following means[25]:

- Paragraph level elements such as **<p>**, **<verbatim>** which is preformatted text.
- Different types of lists, in particular ordered and unordered lists **<list>**, lists with arbitrary labels **<labeled-list>** and definitions **<def-list>**.
- Tables which follow the *CALS* table model **<table>**. This very powerful model allows to merge cells, to define table heads, to define variable column width.
- Cross-references (**<xref>**)
- User defined chapters (**<chapter>**, **<topic>**)
- Figures (**<figure>**) which must be provided as external files in encapsulated postscript format.
- Several character level elements (**<tt>**, **<e>**) which allow to emphasize parts of the text resp. to mark it up as a specific semantic.

[24] we admit that this name is not self explanatory

[25] These means go far beyond the capabilities of HTML. They represent a structure which is strong enough to express most of the technical documents while being simple enough for automatic rendering.

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

- Equations and math formula (**<formula>**) which can be expressed as preformatted text as well as as *TeX* math.

In all specific elements within the *MSRSW.DTD* it is possible to use **<add-spec>** resp. **<add-info>** to add further information as prose description.

The capabilities of **<ncoi>** can be seen in the document you are reading, which is written using the *MSRREP.DTD*.

Further information can be found in *[External Document: Concepts of the MSR application profile V1.x.x / Date: 7.1.1998 / Publisher: MSR-MEDOC Taskforce DTD / URL:  / Relevant Position: ]*

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# App. A Glossary

**ABUS** Automobile Bit-serial Universal Interface

**ADC** Analog to Digital Converter

**AS** Application system

**ASAP** originally ”Arbeitskreis für Standardisierung in der Applikation”, Applications Systems Standardization Working Group.

**ASAM** **A**ssociation for **S**tandardisation of **A**utomation- and **M**easuring Systems

**AuSy** Automation System

**CAN** Controller Area Network

**D-BUS** Diagnostic Bus

**DCI** Document Content Information

**Document Content Information** An approach within MSR to give more precise definitions how to use a DTD in a specific case. For example, there is the ASAM-MC usecase. In addition to this, the DCI allows to specify datatypes, specific instances and restrictions which cannot be expressed using the SGML DTD Syntax.

**DTD** Document type definition - a method to predefine structures of documents or grammars of documents.

**ECU** Electronic Control Unit

**Entity** An entity is a part of an SGML instance representing specific chunks of information kept in separate memory units (files) or in specific data formats.

Entities are also used to specify characters which are not allowed in the text portion of an SGML file (for example ”<” is represented as ”<”).

**FMEA** Failure Mode and Effect Analysis

**HTML** Hypertext Markup Language

**HW** Hardware

**Hytime** Hypermedia/Time-based Structuring Language (ISO 10744)

**Markup** Indication of the meaning of a particular piece of information. In *SGML* markup is specified as a pair of tags enclosing the particular piece of information.

**MCD-System** Measurement Calibration Diagnosis System

**MEDOC** MSR development documentation

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

**MMX** Metamorphosis scripting language

**model group** A group of elements or model groups representing a subgrammar. For example in ”( a, (b | c), d)”, the term ”(b | c)” represents a modelgroup.

**MSR** Manufacturer Supplier Relationship

**SGML** Standard generalized markup language

**SGML instance** A particular information tree in SGML format. This instance can be distributed across multiple files and storage media.

**SW** Software

**W3C** World Wide Web Consortium - a standards body defining recommendations for the World Wide Web (http://www.w3c.org). In particular all XML related Standards are defined there.

**XML** Extensible Markup Language

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# App. B Merged Help set

This chapter is used by the help generator to establish a merged help set.

*[External Document: / URL: msrsw_v220-eadoc.chm]*

*[External Document: / URL: msrsw-tr-chg.sgm.ncoi.chm]*

*[External Document: / URL: msr-tr-cap.chm]*

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# Documentadministration

## Overview of Changes

| Total | Documentpart | Nr. | Change | Reason | Related to |
|---|---|---|---|---|---|
| 23.12.1999 | | 4 | initial draft | | Document |
| 18.1.2000 | | 3 | First responses | | Document |
| 4.4.2001 | | 2 | partly adapt to final version | ASAM release workshop | Document |
| 8.4.2001 | | 1 | continue adapt to final version | post ASAM release workshop | Content |

## Versions Overview

| Document Part | Date | Editor | | | |
|---|---|---|---|---|---|
| | | Company | Version | State | Remarks |
| From page 8 | 2.1<br>WD | Bernhard Weichel | | | |
| | 8.4.2001<br>Changes 1 | ASAM-MSR | 2.1 | Release Doc | |
| | 2.0<br>WD | Bernhard Weichel | | | |
| | 4.4.2001<br>Changes 2 | ASAM-MSR | 2.0 | | |
| | 18.1.2000 | Bernhard Weichel | | | |
| | Changes 3 | ASAM-MSR | 1.2 | WD | |
| | 23.12.1999 | Bernhard Weichel | | | |
| | Changes 4 | ASAM-MSR | 1.1 | CD | |

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# References

## External Documents

| | | |
|---|---|---|
| **Designation:** | ASAP 1b specification | |
| **URL:** | | |
| **Relevant Position:** | Chapter 2.1.9 | 54 |

| | | |
|---|---|---|
| **Designation:** | Concepts of the MSR application profile V1.x.x | |
| **Date:** | 7.1.1998 | |
| **Publisher:** | MSR-MEDOC Taskforce DTD | |
| **URL:** | | |
| **Relevant Position:** | all chapters | 78 |

| | | |
|---|---|---|
| **Designation:** | | 81 |
| **URL:** | msr-tr-cap.chm | |

| | | |
|---|---|---|
| **Designation:** | | 81 |
| **URL:** | msrsw-tr-chg.sgm.ncoi.chm | |

| | | |
|---|---|---|
| **Designation:** | | 81 |
| **URL:** | msrsw_v220-eadoc.chm | |

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# Technical Terms

## Code



## Organisations



## OTHER



## Products



All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

## SGML Attributes



## SGML Elements



All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

## Tools



All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.

# Configuration Parameters

**Company (—company)**
asam-msr

**Language (—lang)**
English

**Treatment of content for Xrefs (—xrefcontent)**
Xref classes are shown

**Specifying ”See” for XRefs**
”See” is to be inserted for xrefs

**Treatment of filenames in graphics (—figname)**
Filenames for graphics are shown

**Treatment of width and height attributes of graphics (—figdimension)**
Width and height of graphics is not considered

**Titlepage Graphic (—graphic)**
No title graphic specified

**Logo Graphic (—head-logo)**
asam+msr-logo.eps

**Fixtext File (—fixtext)**
C:\MSRMETAPAGE\mmapps\msrrep\lib\msrrep_ft.xml

**Output of Local Administrative Data (—admindata)**
Local administrative data is output

**Filename**
C:\MEDOC\msr-docs.new\2001\msrsw-tr-intro\msrsw-tr-intro.sgm

**MetaMorphosis-Version**
3.2

**Form Version**
2.0 (MetaPage)

**Date**
19-Sep-1999

All rights by ASAM-MCD and MSRSW joint taskforce. We reserve all rights of disposal such as copying and passing on to third parties.