


# **MSR-Teilprojekt MEDOC**

## **Strukturelle Grundlagen**


**Bereich: Netzwerke**

**MSR MEDOC, Dipl.-Ing. R. Reimer**

	Strukturelle Grundlagen SNT  Zusammenfassung	Seite: 2/75 Datum: 2002-02-07 Status: RD
---	---	--


## Zusammenfassung

Dieses Dokument beschreibt die strukturellen Grundlagen der MSR-Entwicklungsdokumentation MEDOC für den Bereich *Netzwerke*. Die Beschreibung kann auch als Anleitung zur Erstellung einer strukturierten Entwicklungsdokumentation verwendet werden.


	Strukturelle Grundlagen SNT	Seite: 3/75
	Kapitel: Inhaltsverzeichnis	Datum: 2002-02-07 Status: RD

# Inhaltsverzeichnis


	Inhaltsverzeichnis	3
	Abbildungsverzeichnis	6
	Tabellenverzeichnis	7
	Präliminarien	8
1	<b>Einleitung</b>	<b>9</b>
2	<b>Ziel und Vorgehen</b>	<b>10</b>
1	<b>Allgemeines</b>	<b>11</b>
2	<b>Zuordnung von Netz und Steuergerät</b>	<b>13</b>
2.1	Anforderungen	13
2.2	Beschreibung von Netzwerk-Anschlüssen in MSRDOC	14
3	<b>Strukturierung der MSRNET.DTD</b>	<b>18</b>
3.1	Allgemeine Beschreibung	19
3.2	Netz-Architektur	19
3.2.1	Verbindungskomponenten	19
3.2.2	Topologie	20
3.2.2.1	Netzleitungen	21
3.2.2.2	Beschreibung der Netzknoten	21
3.2.2.3	Beschreibung der Segmentierung	22
3.2.2.4	Beispiel für Beschreibung der Netzwerk - Topologie	22
3.2.3	Beschreibung der Netzwerk - Schnittstellen	24
3.3	Netzwerkbetrieb	26
3.3.1	Allgemeines Netzwerkmanagement	26
3.3.2	Initialisierung	27
3.3.3	Fehlerbehandlung	27
3.3.4	Diagnose	27
3.3.5	Blockübertragungsmodi	27
3.3.6	Netz-Signale	27
3.3.7	Botschaften	29
3.4	Varianten	34

	Strukturelle Grundlagen SNT	Seite: 4/75
	Kapitel: Inhaltsverzeichnis	Datum: 2002-02-07 Status: RD

<b>4</b>	<b>Übersicht Changes</b>	<b>35</b>
4.1	Übersicht zu Zweite Version mit neuem Anwedungsprofil	35
4.1.1	Release notes	35
4.1.1.1	[cmt-text] sw-compu-method-text mit Beschreibung	35
4.1.1.2	[resp.obj] Zuweisung von Verantwortlichkeiten zu Objekten	35
4.2	Übersicht zu Erste Version mit Bugfixes	35
4.2.1	Release notes	36
4.2.1.1	enh: [trans-mode] Transmission-Modes ausmodellieren	36
4.2.1.2	[basetype] sw-base-type in net-signal	36
4.2.1.3	enh: [docrefopt] Dokumentübergreifende Verweise optional	37
4.2.1.4	enh: [init-opt] init-value optional	37
4.2.1.5	enh: [msg-set] Botschaften in einzelne Kapitel	37
4.2.1.6	enh: [msgid-mask] Gleiche Botschaft aus verschiedenen Sendern	37
4.3	Übersicht zu Erste Version	37
4.3.1	Release notes	38
4.3.1.1	enh: [ctrl-mode ] Wo wird der Mode der Controller festgelegt	38
4.3.1.2	bug: [var-trans] Varianten in transmission-spec	38
4.4	Übersicht zu Einführung der Net-Ports	38
4.5	Übersicht zu Sitzung 22.8.96	38
4.6	Übersicht zu Sitzung 12.12.96	39
<b>5</b>	<b>Changes</b>	<b>41</b>
5.1	[gateway] Wie wird Gateway-Funktionalität behandelt	41
5.2	[multi-arch] Unterstützung von Multi-Architektur-Netzen	41
5.3	[net-class] Netzwerk - Protokollklassen	42
5.4	[non-can] Ausdehnung auf nicht CAN-Netze	42
5.5	[supersignal] Supersignal	42
5.6	[sw-var-abgl ] Abgleich zwischen MSRSW-Variablen und MSRNET Signalen	43
5.7	[trans-mode] Transmission-Modes ausmodellieren	43
5.8	[basetype] sw-base-type in net-signal	44
5.9	[cmt-text] sw-compu-method-text mit Beschreibung	44
5.10	[ctrl-mode ] Wo wird der Mode der Controller festgelegt	45
5.11	[docrefopt] Dokumentübergreifende Verweise optional	46
5.12	[init-opt] init-value optional	46
5.13	[msg-set] Botschaften in einzelne Kapitel	47
5.14	[msgid-mask] Gleiche Botschaft aus verschiedenen Sendern	47

	Strukturelle Grundlagen SNT	Seite: 5/75
	Kapitel: Inhaltsverzeichnis	Datum: 2002-02-07 Status: RD

5.15	[ Netport-Node] Behandlung von Knoten mit mehreren Netports zu unflexibel	48
5.16	[resp.obj] Zuweisung von Verantwortlichkeiten zu Objekten	49
5.17	[seg.na] Segmentierung wird nicht immer eingegeben	49
5.18	[var-trans] Varianten in transmission-spec	50
5.19	[Ink-sw ] Wie werden Signale zu SW-Variablen zugeordnet	50
Anh. A	<b>Glossar</b>	<b>52</b>
Anh. B	<b>Literaturverzeichnis</b>	<b>55</b>
Anh. C	<b>Legende der Strukturgrafiken</b>	<b>56</b>
Anh. D	<b>Betrachtete Alternativen zur Ankopplung von SG und Netz</b>	<b>57</b>
Anh. D.1	Kopplung über Einzelsignale	57
Anh. D.2	Verwendung von Namenskonventionen	57
Anh. D.3	Einführung eines Netz-Signalelementes	57
Anh. D.4	Referenzieren von Netzelementen bei der Signaldefinition	58
Anh. E	<b>Festlegung von Wertebereichen</b>	<b>60</b>
Anh. E.1	node-type	60
Anh. E.2	topology-type	60
Anh. E.3	net-signal-class	60
Anh. F	<b>Zusätzliche Anmerkungen zum Dokumentstand 1.1.1 am 10.7.98</b>	<b>61</b>
	Dokumentverwaltung	62
	Literaturverzeichnis	68
	Index	70
	Technische Begriffe	71

	Strukturelle Grundlagen SNT	Seite: 6/75
	Kapitel: Inhaltsverzeichnis	Datum: 2002-02-07 Status: RD

## Abbildungsverzeichnis

Abbildung 1	MSRDOC.DTD und MSRNET.DTD	11
Abbildung 2	Kopplung von Steuergerät und Netz	14
Abbildung 3	Netzwerk-Anschlüsse in MSRDOC	15
Abbildung 4	Beispielszenario	16
Abbildung 5	Grundsätzlicher Aufbau der MSRNET.DTD	18
Abbildung 6	Schichtenmodell für Zeitangaben/Übertragungszeiten	19
Abbildung 7	Struktur für Verbindungskomponenten	20
Abbildung 8	Struktur der Topologie	21
Abbildung 9	Strukture des Netzinterfaces	24
Abbildung 10	Strukturierung des Netzwerkbetrieb	26
Abbildung 11	Netz-Signale	28
Abbildung 12	Struktur der Botschaften	30
Abbildung 13	Aufbau der signale in einer Botschaft	34
Abbildung 14	Bedeutung der Symbole in den Strukturgrafiken	56
Abbildung 15	Zuordnung der Netzleitungen zum Steuergerät (SG)	57

## Tabellenverzeichnis

Tabelle 1	Abgrenzung der MEDOC-Aktivitäten von den Normungsaktivitäten	9
Tabelle 2	Zusammenstellung von Net-Port, Net-Line, und Signalname	17
Tabelle 3	Lösungen in Zweite Version mit neuem Anwedungsprofil	35
Tabelle 4	Requests zu Erste Version mit Bugfixes	36
Tabelle 5	Lösungen in Erste Version mit Bugfixes	36
Tabelle 6	Requests zu Erste Version	37
Tabelle 7	Lösungen in Erste Version	38
Tabelle 8	Requests zu Einführung der Net-Ports	38
Tabelle 9	Lösungen in Sitzung 22.8.96	39
Tabelle 10	Requests zu Sitzung 12.12.96	39

## Präliminarien

Projektbeteiligte  
Firmen

### MSR MEDOC [MEDOC]

Name Rollen	Abteilung	Adresse	Kontakt
Dipl.-Inform. H. Gengenbach			
Dipl.-Ing. E. Hinderer			
Dipl. Ing. E. Jakobi			
Dipl.-Math. M. Krause			
Dipl.-Ing. R. Mertl			
Dipl.-Ing. D. Neumann			
Dipl.-Inform. P. Raulder			
Dipl.-Ing. T. Riegraf			
Dipl.-Ing. B. Weichel			
Dipl.-Inform. J. Wieland			
Dipl.-Ing. R. Reimer			

Versionsinformation

Dokumentteil	Herausgeber			
	Firma	Version	Status	Anmerkungen
2002-02-07	Dipl.-Ing. R. Reimer			
<a href="#">Details siehe Nr. 1, Seite 66</a>	MEDOC	1.1.2	RD	



# 1 Einleitung

Dieses Dokument beschreibt die strukturellen Grundlagen der MSR-Entwicklungsdokumentation MEDOC für den Bereich *Netzwerke*. Die Beschreibung kann auch als Anleitung zur Erstellung einer strukturierten Entwicklungsdokumentation verwendet werden.

Die Beschreibung erfolgt u.a. in Strukturgrafiken, die aus der prototyp-Implementierung der *MSRNET.DTD* gewonnen wurden. Die Bedeutung der Symbole ergibt sich aus [Abbildung 14 Bedeutung der Symbole in den Strukturgrafiken S. 56](#).


An dieser Stelle wird daraufhingewiesen, daß MSR keine Normung betreibt und auch in Zukunft keine betreiben wird. In der Anforderungsdokumentation und in der Entwicklungsdokumentation MEDOC werden lediglich Inhalt und Struktur von bestehenden Normen/Hausnormen verwendet. MEDOC soll die Möglichkeit bieten, die Inhalte der (inter-)nationalen Normen und Hausnormen sowie nicht genormte, für die Beschreibung von Systemen relevante Daten in einer Entwicklungsdokumentation abzulegen.

Der vereinfachte Sachverhalt ist in der nachfolgenden Tabelle dargestellt:

**Tabelle 1: Abgrenzung der MEDOC-Aktivitäten von den Normungsaktivitäten**

MEDOC	Norm	Hausnorm	keine Norm
Struktur	Struktur der Normen kann von MEDOC abgebildet werden	Struktur der Hausnormen kann abgebildet werden	Vom MSR-Kreis abgestimmt
Inhalt	Stammdaten aus Normen übernommen	Stammdaten können spezifisch ergänzt werden	

MEDOC erlaubt es, die Inhalte und die Struktur der internationalen, nationalen und der Hausnormen aufzunehmen. Grundlage für die Strukturdefinitionen sind die zum Zeitpunkt der Drucklegung gültigen Normen. Darüber hinaus können Datenbestände, die durch keine Norm abgebildet werden, mittels MEDOC erfaßt werden. Die Struktur der Daten wird vom MSR-Kreis abgestimmt, die Inhalte werden keinerlei Normung unterworfen. Tabellen und Tabelleninhalte, die sich in diesem Dokument befinden, sind nur als Beispieldaten anzusehen. Da für die einzelnen Konsortiumsfirmen unterschiedliche Grenzwerte gelten, wurden die Inhalte z.T. den internationalen Normen entnommen.

	Strukturelle Grundlagen SNT Kapitel: Ziel und Vorgehen	Seite: 10/75 Datum: 2002-02-07 Status: RD
---	--	---

## 2 Ziel und Vorgehen

Ziel der Arbeitsgruppe *MEDOC-Netzwerke* ist es, ein Format für alle möglichen Busbeschreibungen bzw. Beschreibungen von Netzwerken zu definieren. Die Datenmodellierung wird zunächst anhand des *CAN* vorgenommen (Full-CAN und Basic-CAN). Nach Abschluß der Datenmodellierung wird die Übertragung auf weitere Netz-Formate vorgenommen.

# 1 Allgemeines

Die bislang modellierte *MSRDOC.DTD* bildet in einer Instanz stets ein System inklusive der zugehörigen *Steuergeräte* bzw. des zugehörigen *Steuergerätes* ab (z.B. ein ABS-System). Da ein Netz i.a. mehrere *Steuergeräte* in unterschiedlichen Systemen verbindet (z.B. das *Motor-Steuergerät* und das *ABS-Steuergerät*), ist es sinnvoll, die Beschreibung von Netzwerken in einer eigenständigen DTD abzubilden.

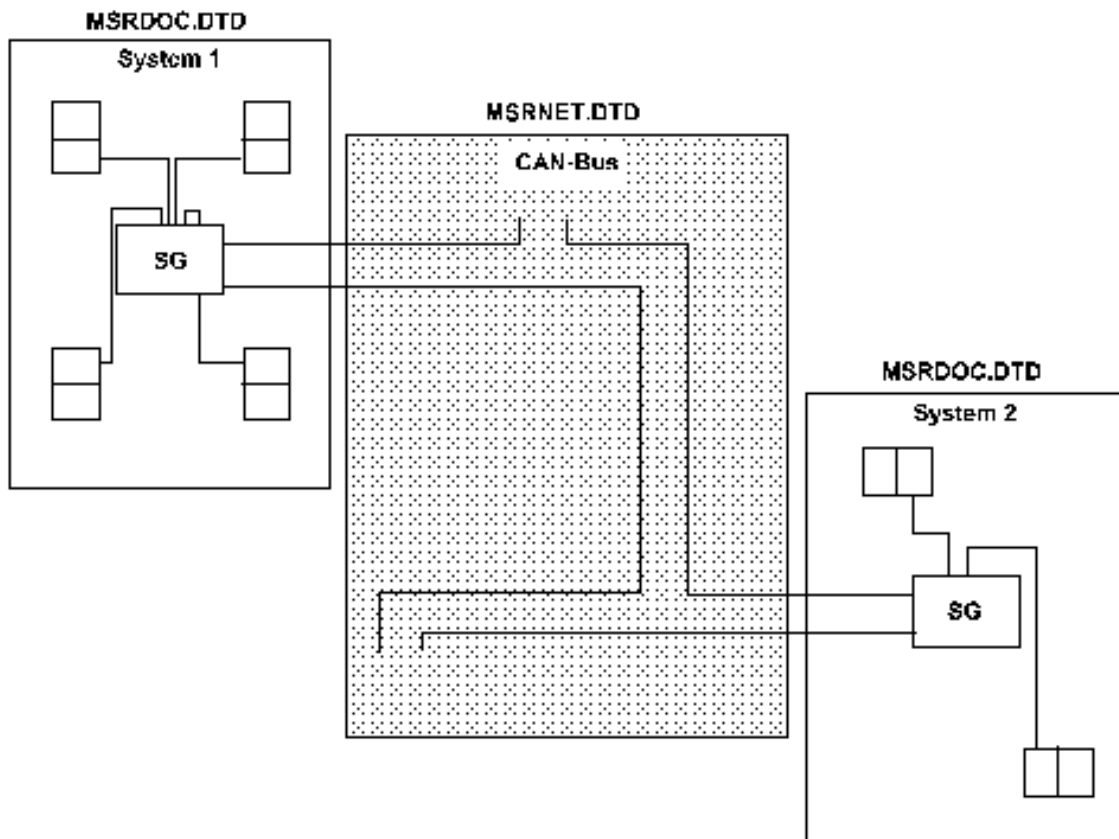



Abbildung 1: MSRDOC.DTD und MSRNET.DTD

Für die Beschreibung der Ankopplung der Steuergeräte an das Netz sind instanzübergreifende Links notwendig. In Anlehnung an die *MSRDOC.DTD* ist ein entsprechendes Variantenkonzept vorgesehen, so daß die Bezüge zwischen Instanzen der *MSRDOC.DTD* und der *MSRNET.DTD* variantenbezogen beschrieben werden können. Die baureihenbezogene oder baureihenübergreifende Beschreibung von Netzen muß möglich sein.

	<p style="text-align: center;">Strukturelle Grundlagen SNT</p> <p>Kapitel:           Allgemeines</p>	<p>Seite:     12/75</p> <p>Datum:    2002-02-07</p> <p>Status:    RD</p>
---	--	--


Besitzt ein *Fahrzeugsystem* mehrere Netzsysteme parallel, so sind diese in entsprechend vielen Instanzen der *MSRNET.DTD* zu beschreiben. Damit wird eine klare Trennung (technisch wie auch organisatorisch) der verschiedenen Netzen erreicht<sup>1</sup>.

Für Simulationszwecke erhält der Zulieferer vom Hersteller die Beschreibung der gesamten Netztopologie in den betroffenen Baureihen.

Eine Verknüpfung zwischen der *MSRNET.DTD* und den *MSRDOC.DTDs* ist über die Kurzbezeichnung der Netzsignale denkbar.

---

<sup>1</sup> Gateways legen zunächst eine Zusammenfassung mehrerer Netze in einer Instanz nahe. Die Gateway-Funktionalität wird aber in einem konkreten System bzw. Gerät realisiert, und ist daher in dessen Verhalten bzw. Software zu beschreiben.

	Strukturelle Grundlagen SNT	Seite: 13/75
	Kapitel: Anforderungen	Datum: 2002-02-07 Status: RD

## 2 Zuordnung von Netz und Steuergerät

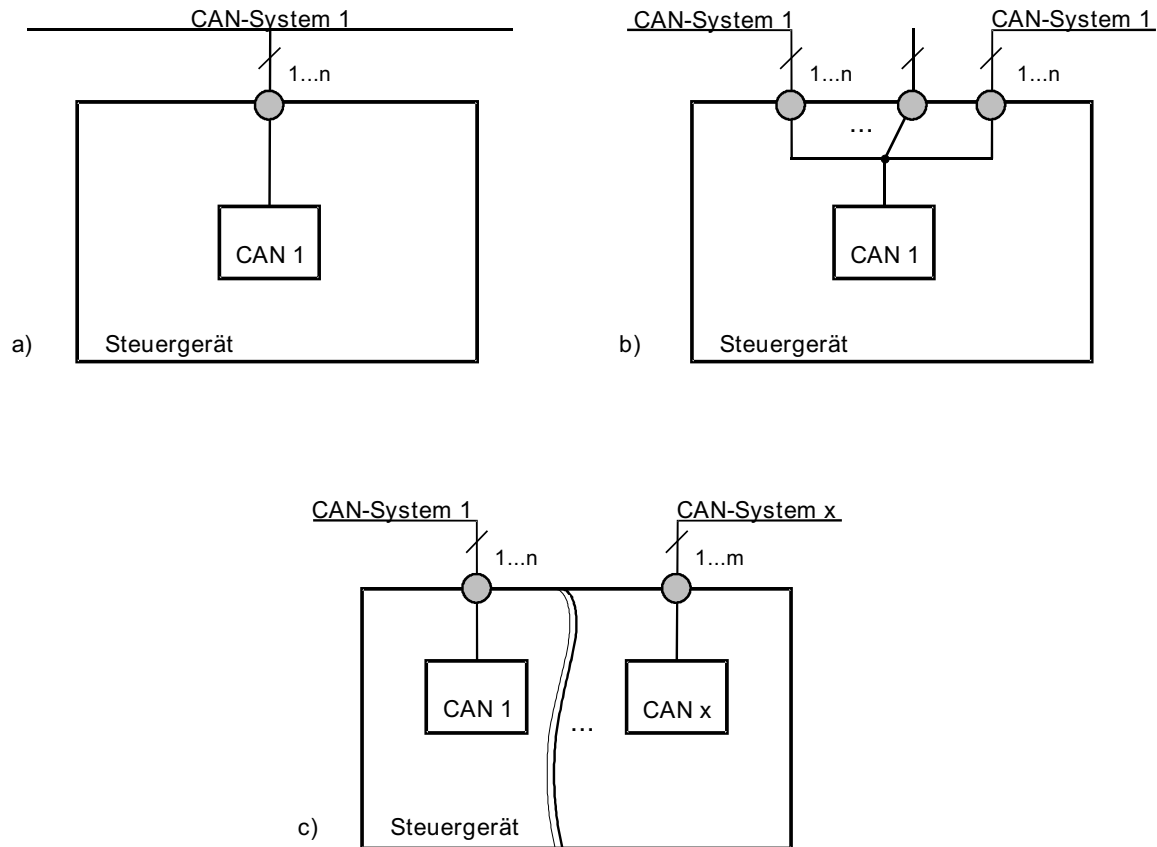
### 2.1 Anforderungen

Die Kopplung eines *Steuergerätes* mit einem oder mehreren Netzen kann auf unterschiedliche Art und Weise erfolgen. In [Abbildung 2 Kopplung von Steuergerät und Netz S. 14](#) sind drei dieser Möglichkeiten aufgezeigt:

- a) Stichleitung und Netzsystem
- b) Abzweig bzw. Sternpunkt im *Steuergerät* (mehrfach n-adrige Leitungen)
- c) Verbindung zu mehreren Netzen (z. B. Gateway)

Allen drei Ansätzen ist gemeinsam, daß die Zuordnung auf Basis von Steuergeräte-Signalen zu Namen von Netzleitungen erfolgt<sup>2</sup>. Eine Zuordnung zu Steuergeräte-Port erscheint nicht flexibel genug im Hinblick auf die Verwendung einer Netz-Beschreibung für mehrere Steuergerätfamilien.

<sup>2</sup> Hier wird nicht der Begriff Netz-Signal verwendet, da dieser im Zusammenhang mit den Botschaften in anderer Bedeutung wieder auftritt.



**Abbildung 2: Kopplung von Steuergerät und Netz**

Während in Teilabbildung a) das *Steuergerät* über eine n-adrige Stichleitung (z.B. CAN\_High, CAN\_Low, CAN\_Shield) mit dem Netz verbunden ist, sind bei b) diese n-adrigen Signalleitungen mehrfach vorhanden. Im Fall b) befindet sich im *Steuergerät* ein Abzweig oder Sternpunkt des Netzes.

Weiterhin kann ein *Steuergerät*, wie in Teilabbildung c), mit verschiedenen Netzen verbunden sein (z.B. Gateway). Dabei sind die Ausführungen der Schnittstellen zu den jeweiligen Netzen nach a) und b) möglich.

Für die Beschreibung der Kopplung von *Steuergerät* und Netz wurden neben der gewählten Lösung auch die Alternativen in [Topic Anh. D Betrachtete Alternativen zur Ankopplung von SG und Netz S. 57](#) betrachtet. Es wird die Lösung in [Topic 2.2 Beschreibung von Netzwerk-Anschlüssen in MSRDOC S. 14](#)) eingesetzt.

## 2.2

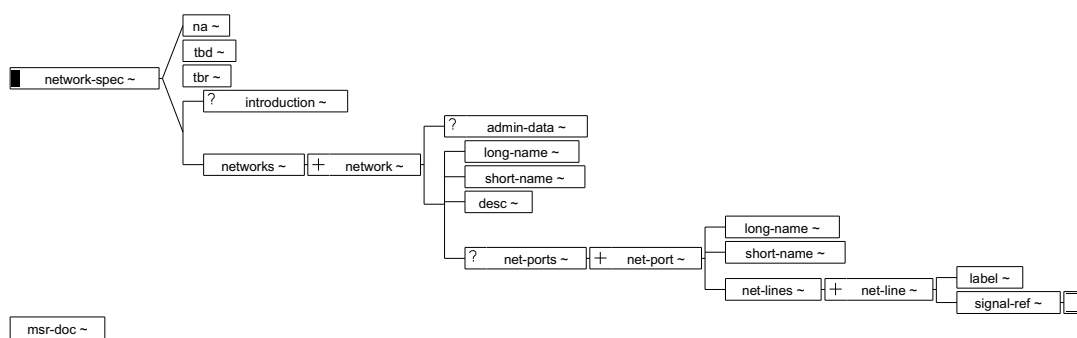
## Beschreibung von Netzwerk-Anschlüssen in MSR-DOC

Unter der `<architecture>` in `MSRDOC.DTD` wird ein neues Element `<network-spec>` (s.[Abbildung 3 Netzwerk-Anschlüsse in MSRDOC S. 15](#)) eingeführt, um die Netzankopp-

lungen des Bauteiltyps definieren zu können. Diese Modellierung wurde aus folgenden Gründen gewählt:

- Die Ziele für eine (semantisch auszuführende) Referenzierung aus *MSRNET.DTD* Instanzen werden in **<network-spec>** konzentriert.
- Netzbezogene Eigenschaften und Zusammenhänge von Signalen können hier konzentriert werden, ohne das Inhaltsmodell von **<signal>** generell auszuweiten und gleichzeitig Netzsignale wie alle anderen Signale behandeln zu können.

Eine mögliche semantische Überschneidung mit **<signal-class>** wird bewußt in Kauf genommen, da letzere für firmen- bzw. prozeßspezifische Klassifizierungen konzipiert ist.



**Abbildung 3: Netzwerk-Anschlüsse in MSRDOC**

### Beispiel der Eingabe in MSRDOC - Instanz

Eine *MSRDOC.DTD*-Instanz (für Steuergerät 2 in [Abbildung 4 Beispielszenario S. 16](#)) sieht im Prinzip wie folgt aus (das Gegenstück in der *MSRNET.DTD*-Instanz ist in [Topic 3.2.2.4 Beispiel für Beschreibung der Netzwerk - Topologie S. 22](#) dargestellt):

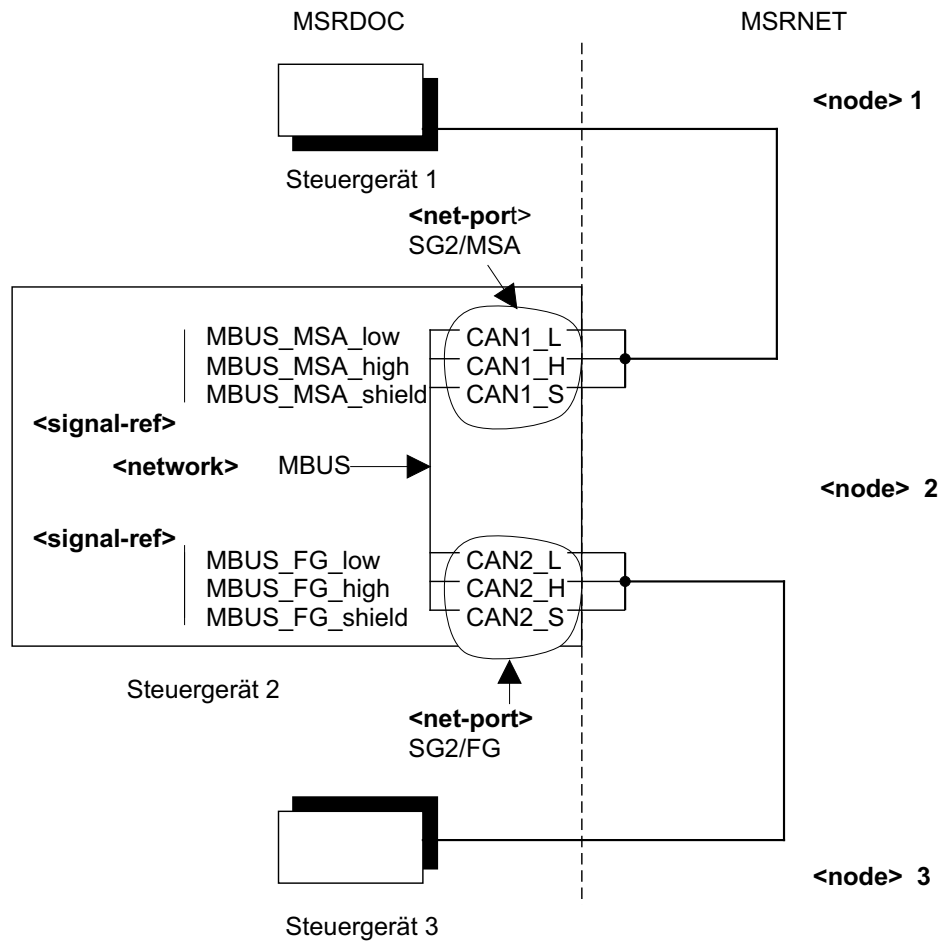


Abbildung 4: Beispielszenario

```

<part-type>
  <long-name>Steuergerät 2</long-name>
  <short-name>SG2</short-name>
  ...

<architecture>
  <scheme-diagrams> ...
  <interface-spec> ...
  <signal-spec> ...
  <network-spec>
    <networks>
      <network><long-name>CAN-Bus der Motorsteuerung</>
      <short-name>MBUS</>
      <desc>Dieser Bus behandelt primär die Motorsteuerung</>
      <net-ports>
        <net-port>
          <long-name>motorraumseitiger Anschluß</>
          <short-name>MSA</>
          <net-lines>
            <net-line>
              <net-line-name>CAN_LOW</>
              <signal-ref sref="MBUS_MSA_LOW">
            </>
            <net-line>
              <net-line-name>CAN_High</>
              <signal-ref sref="MBUS_MSA_HIGH">
            </>
          </>
        </>
      </>
    </>
  </>

```



```

<net-line>
  <net-line-name>CAN_shield</>
  <signal-ref sref="MBUS_MSA_SHIELD">
</>
</net-port>
<net-port>
<long-name>fahrgeraestraumseitiger Anschluß</>
<short-name>FG</>
<net-lines>
  <net-line>
    <net-line-name>CAN_LOW</>
    <signal-ref sref="MBUS_FG_LOW">
  </>
  <net-line>
    <net-line-name>CAN_HIGH</>
    <signal-ref sref="MBUS_FG_HIGH">
  </>
  <net-line>
    <net-line-name>CAN_SHIELD</>
    <signal-ref sref="MBUS_FG_SHIELD">
  </>
</net-port>
</net-ports>
</network>
</networks>
</network-spec>

...
</part-type>

```

**Tabelle 2: Zusammenstellung von Net-Port, Net-Line, und Signalname**

Net-Port	Net-Line	Signal
MSA	CAN_LOW	MBUS_MSA_LOW
	CAN_HIGH	MBUS_MSA_HIGH
	CAN_SHIELD	MBUS_MSA_SHIELD
FG	CAN_LOW	MBUS_FG_LOW
	CAN_HIGH	MBUS_FG_HIGH
	CAN_SHIELD	MBUS_FG_SHIELD

Diese Variante modelliert die komponentenspezifischen Netz-Eigenschaften komplett. Dazu gehört auch die explizite Zuordnung von Komponenten-Signalname und Netzleitung<sup>3</sup>.

Netz-Ports werden eingeführt, um [Abbildung 2 Kopplung von Steuergerät und Netz S. 14](#) Fall b) abdecken zu können. In diesem Fall liegt auf einer Komponente das Netz mehrfach (also auf mehreren Ports) auf und wird innerhalb der Komponente verbunden. Der Sternpunkt liegt also innerhalb der Komponente. Dabei ist davon auszugehen, daß die Netzleitungen an den jeweiligen Pins unterschiedlich benannt werden. Daher muß eine Netz-Port spezifische Zuordnung von Netzleitungsname zu komponenteninternen Signalnamen erfolgen. Diese Zuordnung kann im Prinzip in zwei Varianten erfolgen:

- Zuordnung von Netzleitung und Komponenten-Signalname pro Netz-Port
- Zuordnung von Netz-Port und Komponenten-Signalname pro Netzleitung<sup>4</sup>
- 

<sup>3</sup> Die Netzleitungs-Bezeichner werden nicht explizit normiert. Es ist Aufgabe einer semantischen Prüfung sicherzustellen, daß die Bezeichner der Netzleitungen der verschiedenen Net-Ports konsistent sind.

<sup>4</sup> Diese Lösung wurde nicht gewählt und ist daher auch nicht als Beispiel aufgeführt.

### 3 Strukturierung der MSRNET.DTD

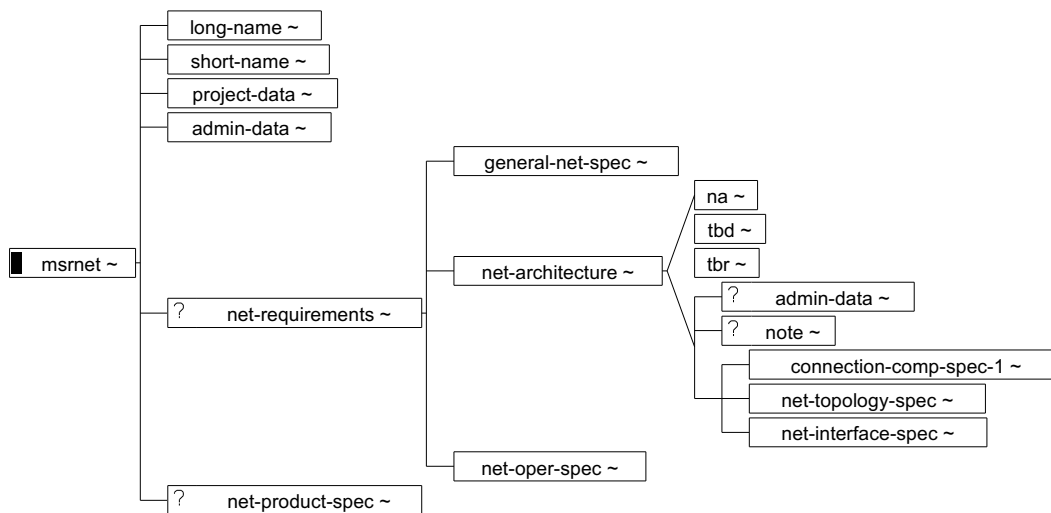
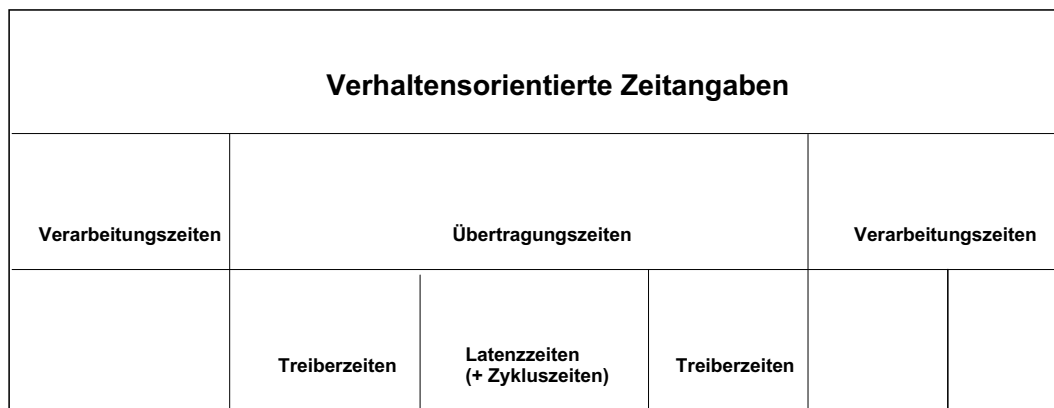


Abbildung 5: Grundsätzlicher Aufbau der MSRNET.DTD

Die Beschreibung eines CAN-Netzwerkes gliedert sich in folgende Teile:

- Lang- und Kurzbezeichnung (<**long-name**> <**short-name**>)
- Verwaltungsinformation < **admin-data**>
- Projektinformationen <**project-data**>
- Allgemeine, globale Beschreibung <**general-net-spec**> [Topic 3.1 Allgemeine Beschreibung S. 19](#)
- Netz-Architektur (<**net-architecture**>) (Topologie, Kabel, Netzparameter, Treiberkonzept, Netz-EMV-Design)
- Netzwerkbetrieb (<**net-oper-spec**>) (Datenverkehr, Signale, Botschaften).

Eine grobe Veranschaulichung in Form eines Schichtenmodelles (ähnlich dem ISO/OSI-Referenzmodell für Kommunikation) bietet die folgende Abbildung:



bus04.gif

**Abbildung 6: Schichtenmodell für Zeitangaben/Übertragungszeiten**

Die erste Version der DTD enthält keine formale Beschreibung von geforderten Übertragungszeiten<sup>5</sup>.

Zunächst sind bei der Botschaftsbeschreibung lediglich die Parameter *Latenzzeit* und *Zykluszeit* vorgesehen, innerhalb der Abbildung also Parameter in der unteren Schicht.

## 3.1 Allgemeine Beschreibung

In **<general-net-spec>** können allgemeine, globale Beschreibungen zu einem Netz abgelegt werden. Darin wird z.B. beschrieben, wofür das Netz eingesetzt wird (Antriebsstrang, Komfortbereich ...).

## 3.2 Netz-Architektur

Die Beschreibung der Netz-Architektur (**<net-architecture>**) teilt sich in die Themen "Verbindungskomponenten" **<connection-comp-spec-1>**, "Topologie" (**<net-topology-spec>**) und "Netz-Interface" **<net-interface-spec>** auf. Hier können sowohl feste wie auch freie Parameter des Netzes bzw. der Netz-Hardware beschrieben werden.

**<add-spec>** dient zur Angabe weiterer, in der DTD nicht direkt vorgesehener Spezifikationen.

### 3.2.1 Verbindungskomponenten

Bei den Verbindungskomponenten werden die verwendeten Netz-Kabel als **<connection-comp-spec-1>** (siehe [Abbildung 7 Struktur für Verbindungskomponenten S. 20](#)) ähnlich wie in *MSRDOC.DTD* beschrieben.<sup>6</sup>

<sup>5</sup> Eine solche Beschreibung formaler Art findet sich nicht in den derzeit vorliegenden Lastenheften.

<sup>6</sup> Es wird eine Abwandlung verwendet, die keinen part-type-ref zulässt.

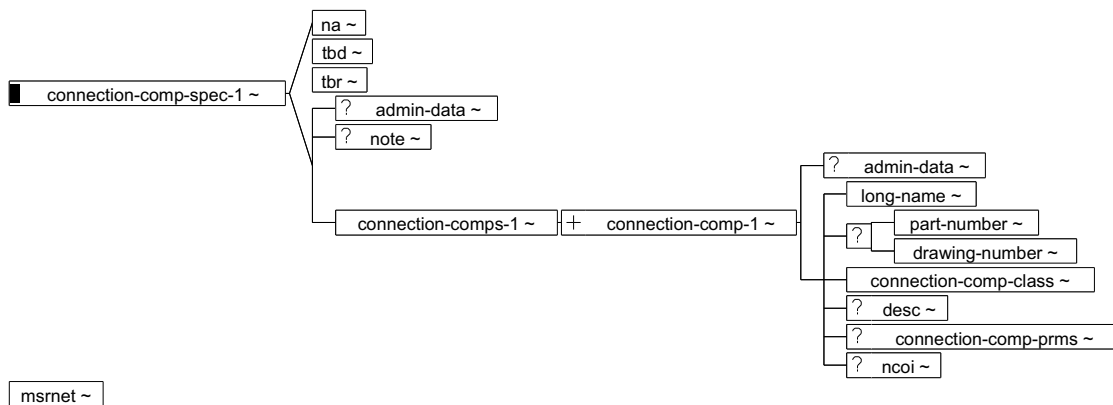


Abbildung 7: Struktur für Verbindungskomponenten

### 3.2.2 Topologie

Innerhalb der Topologie (<net-toplogy-spec > siehe [Abbildung 8 Struktur der Topologie S. 21](#)) können die Art, die Knoten sowie die Segmentierung dokumentiert werden. Die Art (<topology-type>) kann Ausprägungen wie "Ring", "Stern", "Bus", "Bus mit Stichleitung" oder auch "Mischform" besitzen und muß durch eine Grafik ergänzt werden.

Weiterhin werden Knoten <net-nodes> ([Topic 3.2.2.1 Netzleitungen S. 21](#)) und Segmentierung <segmentation> ([Topic 3.2.2.3 Beschreibung der Segmentierung S. 22](#)) beschrieben.

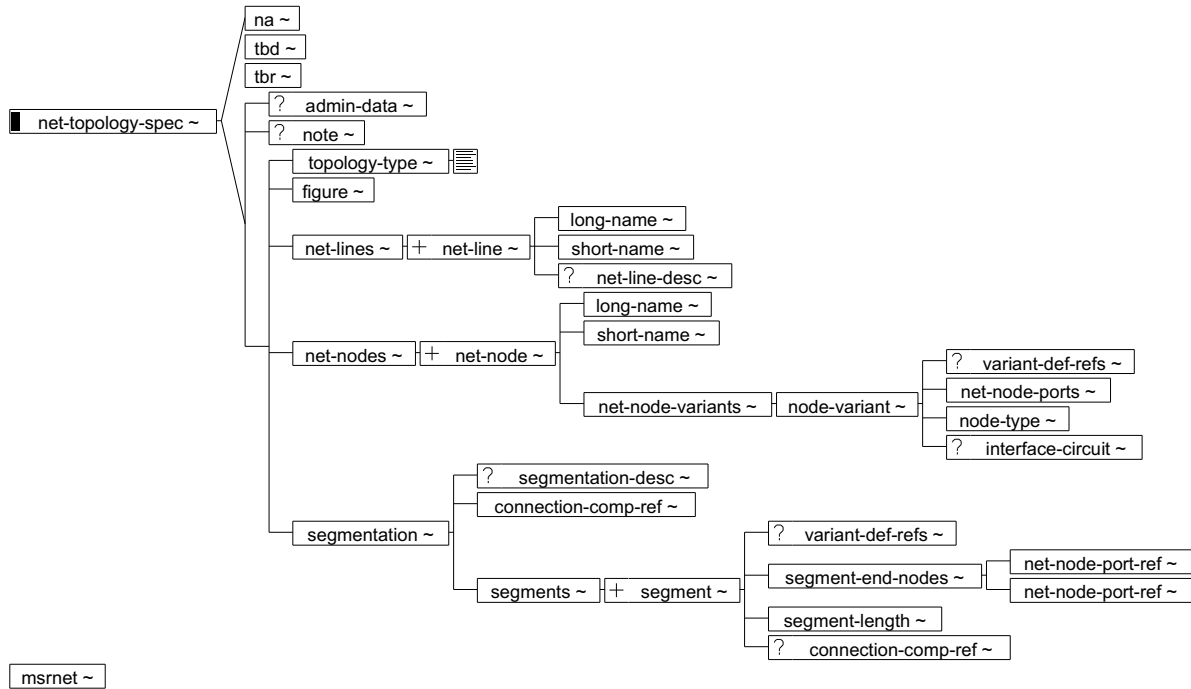


Abbildung 8: Struktur der Topologie

### 3.2.2.1 Netzleitungen

Das Netzverdrahtung selbst besteht in der Regel aus mehreren Leitungen (hier Netzleitung) genannt. Beispiel hierfür ist ( *CAN\_LOW*, *CAN\_HIGH*, *CAN\_SHIELD*). Diese Signale können ebenfalls unter Topologie (<net-lines>) beschrieben werden. Die Beschreibung umfaßt zuminest einen <short-name > und einen <long-name>. Optional kann eine ausführliche Darstellung (<netline-desc>) mit Tabellen, Grafiken usw. gegeben werden .

In den *MSRDOC.DTD* - instanzen werden diese Netzleitungen den komponenteninternen Signalen zugeordnet (vgl [Topic 2.2 Beschreibung von Netzwerk-Anschlüssen in MSRDOC S. 14](#), < net-line>). Damit ist es möglich, die Konsistenz zu überprüfen<sup>7</sup> (Semantik).


Eine explizite Zuordnung dieser Leitungen zu den Adern (z.B. über Farbe) des Verbindungskabels erfolgt nicht, da dieses nicht fomell hierarchisiert beschrieben wird.

### 3.2.2.2 Beschreibung der Netzknoten

Die Beschreibung des Knotens besteht aus:

< short-name> Nummer bzw. Kurz-Bezeichnung des Knotens im Netzwerk

<sup>7</sup> Auf eine formale Referenzierung der netlines zwischen msrdoc und msrnet wurde aus pragmatischen Gründen verzichtet, da i.d.R. in allen Instanzen gleiche Bezeichnungen verwendet werden.

	Strukturelle Grundlagen SNT	Seite: 22/75
	Kapitel: Beispiel für Beschreibung der Netzwerk - Topologie	Datum: 2002-02-07 Status: RD

- < long-name >** Bezeichnung des Knotens im Netzwerk
- <node-type >** Die Knotenart dient zur Unterscheidung von *Teilnehmer* und *Hilfsknoten*. Hilfsknoten nehmen nicht am Netzverkehr teil sondern dienen nur zum physikalischen Aufbau bzw. Darstellung der Topologie.
- < net-node-port >s** Hier werden die Ports des Knotens definiert (vgl. [Topic 2.2 Beschreibung von Netzwerk-Anschlüssen in MSRDOC S. 14](#)). Diese Stelle ist die einzige Stelle, an der referentielle Konsistenz zu *MSRDOC.DTD*-Instanzen gewährleistet werden muß. Das **<net-node-port >** referenziert aus der *MSRDOC.DTD* Instanz das Bauteil (**< part-ref >**) sowie den dazu gehörenden Bauteiltyp enthaltenen Netport (**<net-port-ref >**).<sup>8</sup>
- <interface-circuit >** Angaben zur Abschlußbeschaltung. Hier kann **<interface-circuit >** verwendet werden. Diese Information ist redundant zur *MSRDOC.DTD*. Die Struktur wird vorgehalten, um eine MSRNETH-Instanz autonom beschreiben zu können. Außerdem kann hier die gesamte Abschlußbeschaltung eines Knotens angegeben werden, wohingegen bei *MSRDOC.DTD* nur die Beschaltung eines einzelnen Pins beschrieben ist. Die Angabe ist optional.

### 3.2.2.3 Beschreibung der Segmentierung

Die Segmentierung kann sowohl allgemein (durch Text und Grafik) (**<segmentation-desc >**) dokumentiert werden, als auch formell (**<segments >**) spezifiziert werden.


In **<connection-comp-ref >** kann die standardmäßige Kabelart für das Netz angegeben werden.

Die formelle Spezifikation beschreibt alle Segmente (**<segment >**):

- Segmente können variantenspezifisch angegeben werden (spezifiziert in **<variant-def-refs >**).
- Angaben zu Start-Knoten, Ende-Knoten, Segment-Längen (**<segment-length >**) sowie Kabelarten der einzelnen Segmente. Die Start- und Ende-Knoten werden durch Referenzierung auf einen der **< net-node-port >s** des betroffenen Knotens festgelegt (**< net-node-port-ref >**).
- Als Segment-Länge ist die Angabe in der SI-Einheit Meter vorgesehen<sup>9</sup>.
- Es kann eine segmentspezifische Kabelart angegeben werden (**<connection-comp-ref >**), falls diese von der standardmäßigen Kabelart für das Netz abweicht.

<sup>8</sup> Alle node-ports eines Knotens müssen das selbe part referenzieren. Trotzdem wurde die Referenz auf das Part nicht aus den node-ports "ausgeklammert". Dies hätte bedeutet, daß die Semantik der Referenz verteilt definiert ist. Dies sollte aus Handhabungsgründen vermieden werden. Es ist Aufgabe einer semantischen Prüfung, eventuelle Unstimmigkeiten festzustellen.

<sup>9</sup> Hier wurde auf den Einsatz des Standard-Parameter-Modell verzichtet, um die Darstellung der Segmentierung als geschlossene Tabelle zu ermöglichen.

	Strukturelle Grundlagen SNT	Seite: 23/75
	Kapitel: Beispiel für Beschreibung der Netzwerk - Topologie	Datum: 2002-02-07 Status: RD

### 3.2.2.4 Beispiel für Beschreibung der Netzwerk - Topologie

Das folgende Beispiel<sup>10</sup> entspricht dem Beispiel in [Abbildung 4 Beispielszenario S. 16](#):

```

<net-topology-spec>
  <net-topolgy-type>bus</>
  <connection-comp-ref connection-comp="nyfaz"></>
  <nodes>
    <node>
      <long-name>Steuergerät 1</>
      <short-name>1</>
      <node-variants>
        <node-variant>
          <node-ports>
            <node-port id="sg1.can">
              <long-name>SG1 - CAN</>
              <short-name>can</>
              <net-port-ref netport="SG1/CAN">
            </node-port>
          </node-variant>
        </node-variants>
      <node-type>Teilnehmer</>
    </node>
    <node>
      <long-name>Steuergerät 2</>
      <short-name>SG2</>
      <node-variants>
        <node-variant>
          <node-ports>
            <node-port id="sg2.msa">
              <long-name>Motorraum-CAN</>
              <short-name>MSA</>
              <net-port-ref netport="SG2/MSA">
            </node-port>
            <node-port id="sg2.fg">
              <long-name> Fahrgastraum-CAN</>
              <short-name>FG</>
              <net-port-ref netport="SG2/FG">
            </node-port>
          </node-variant>
        </node-variants>
      <node-type>Teilnehmer</>
    </node>
    <node>
      <long-name>Steuergerät 3</>
      <short-name>3</>
      <node-variants>
        <node-variant>
          <node-ports>
            <node-port id="sg3.can">
              <long-name>Steuergerät3</>
              <short-name>can</>
              <net-port-ref netport="SG3/can">
            </node-port>
          </node-variant>
        </node-variants>
      <node-type>Teilnehmer</>
    </node>
  </nodes>

  <segmentation>
    <segments>
      <segment>
        <segment-end-nodes>
          <node-port-ref node-port="sg1.can"></>
          <node-prot-ref node-port="sg2.msa"></>
        </segment-end-nodes>
        <segment-length>1</>
      </segment>
    </segments>
  </segmentation>

```

<sup>10</sup> Das Beispiel muß noch aktualisiert werden.

```

<segment>
  <segment-end-nodes>
    <node-port-ref node-port="sg2.fg"></>
    <node-prot-ref node-port="sg3.can"></>
  </segment-end-nodes>
  <segment-length>1</>
</segment>
</segments>
</net-topology-spec>

```

### 3.2.3 Beschreibung der Netzwerk - Schnittstellen

Den zweiten Teil der Hardware-Dokumentation bildet das Netz-Interface (**<net-interface-spec>**). Diese beschreibt das Treiberkonzept (**<driver-concept>**), die global für alle Teilnehmer festgelegten Netzparameter (**<net-interface-prms>**), Hinweise zum Netz-EMV-Design (**<net-emc-design>**) sowie ergänzende Angaben (**<add-info >**).

Die Beschreibung des Bus-Interfaces oder auch der Bus-Ankopplung besteht aus einem Bus-globalen physikalischen Datenblatt (requirements); entweder gemäß [Externes Dokument: ISO Norm CAN High-Speed / URL: / relevante Stelle: ] [Externes Dokument: ISO Norm CAN Low-Speed / URL: / relevante Stelle: ] oder oder individuell ausgeprägt. Auf Teilnehmerseite (SG-Seite) ist dann festzulegen, welcher Chip eingesetzt wird (**<part-type-spec >** in MSRDOC).

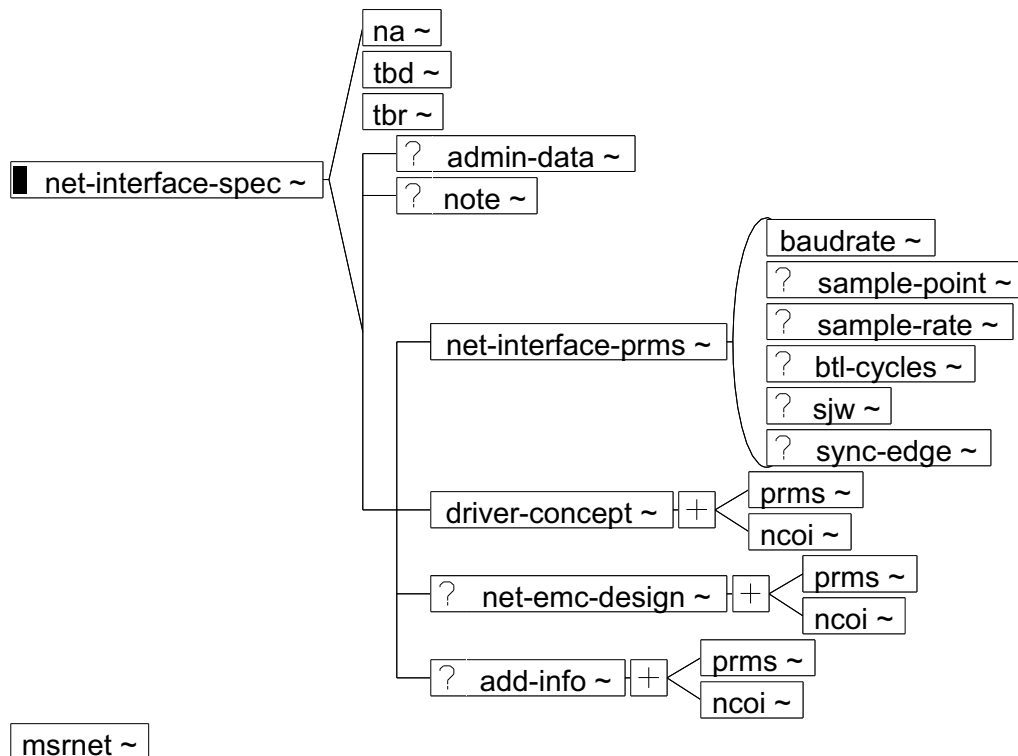



Abbildung 9: Struktur des Netzinterfaces



	Strukturelle Grundlagen SNT	Seite: 25/75
	Kapitel: Beschreibung der Netzwerk - Schnittstellen	Datum: 2002-02-07 Status: RD

Die global für alle Teilnehmer festgelegten Netzparameter werden als Parametertabelle (**<net-interface-prms >**) erfaßt<sup>11</sup>:

- <baudrate>** die Baudrate ([Definition Baudrate S. 52](#)) in Hz. Es muß **<abs>-<tol>** - Modell verwendet werden. Die Toleranz ist in % anzugeben.

Die Baud-Raten-Programmierung für den *CAN-controller* leitet sich aus den obigen Parametern ab, ist allerdings nicht eindeutig (z.B. *Registerbelegung*). Es handelt sich hier um eine Beschreibung der Bus-Ankopplung auf einer höheren Ebene (höher als die physikalische Ebene). Man kann auch von der Parametrisierung des CAN-Controllers sprechen. Diese ist beim CAN-Controller bzw. beim *Steuergerät* zu dokumentieren.
- <sample-point>** der Abtastzeitpunkt eines einzelnen Bits in einer Botschaft. Die Angabe erfolgt in Prozent. (**<abs>-<tol>** - Modell, ohne ausgefüllte Toleranz)

Dies ist eine optionale Angabe.

Dieser Parameter gibt an, zu welchem Zeitpunkt ein Bit abgetastet wird. Er wird in Prozent der Bitzeit (Übertragungszeit für ein Bit) angegeben. Ein typischer Wert ist 75 %.
- <sample-rate>** Anzahl der Abtastungen pro Bit

Dies ist eine optionale Angabe.
- <btl-cycles>** Anzahl der [Definition BTL-Zyklen S. 52](#) BTL-Zyklen als ganzzahligen Wert, d.h. die Auflösung eines Bit (**< abs>-<tol>** - Modell ohne ausgefüllte Toleranz).

Dies ist eine optionale Angabe.
- <sjw>** SJW (Synchronization Jump Width) als ganzzahligen Wert ([Definition SJW S. 54](#)). Der Parameter "SJW" (Syncro Jump Width) wird ebenfalls in Prozent der Bitzeit angegeben.

Dies ist eine optionale Angabe.
- <sync-edge>** Synchronisationsflanke, anzugeben als **<text>** - Parameter

Dies ist eine optionale Angabe.

Die **<baudrate>** ist Bus-global. Die Parameter **<sample-point>**, **<sjw >** und **<btl-cycles>** sind i.A. global, können in Ausnahmefällen aber auch teilnehmerspezifisch sein Diese Sonderfälle sind nicht in *MSRNET.DTD* abgedeckt..

Innerhalb des optionalen Treiberkonzepts (**< driver-concept>**) können die zu verwenden- den bzw. die verwendeten Treiberbausteine beschrieben werden.

Ebenso wie für das Treiberkonzept steht für die Beschreibung des Netz-EMV-Designs eine freie Struktur (**<net-emc-design >**) zur Verfügung.

<sup>11</sup> für jeden dieser Parameter sollte eigentlich ein spezifisches Parametermodell verwendet werden. Im Interesse der Verallgemeinerung wurde jedoch auf das MSR-Standardparametermodell zurückgegriffen. Wenn dieses künftig als Architectural Form ausgeführt wird, könnten spezifischere Modelle verwendet werden. Zunächst muß der Benutzer das richtige Modell wählen, bzw. die Autorenumgebung sollte das richtige Model vorschlagen.

Bei den ergänzenden Angaben (< **add-info** >) sind ggf. weitere physikalische Parameter wie Flankensteilheit, Eingangswiderstand und Signalpegel zu dokumentieren.

### 3.3 Netzwerkbetrieb

Dieser Abschnitt (<**net-oper-spec**>) beschreibt den Datenverkehr für unterschiedliche Netzwerkdienste ([Topic 3.3.1 Allgemeines Netzwerkmanagement S. 26](#) sowie die Verhaltensweisen bei Auftreten von Fehlern ([Topic 3.3.3 Fehlerbehandlung S. 27](#)) u. ä. Die Strukturen der blockweisen Datenübertragung werden in [Topic 3.3.5 Blockübertragungsmodi S. 27](#) dargestellt.

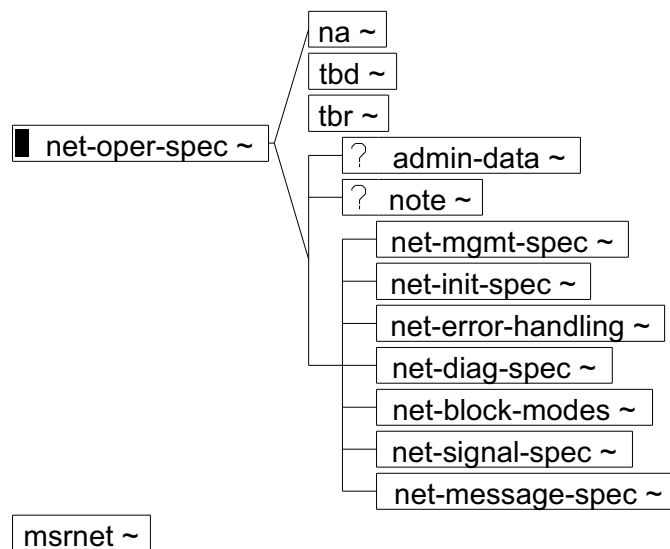



Abbildung 10: Strukturierung des Netzwerkbetrieb

Eine Botschaft, die über CAN verschickt wird, enthält in der Regel mehrere Signale. Dabei ist es möglich, daß ein Signal in mehr als einer Botschaft verwendet wird. Daher werden zunächst die Signale (<**net-signal-spec**>, s. [Topic 3.3.6 Netz-Signale S. 27](#)), und dann die Botschaften (<**net-message-spec**> s. [Topic 3.3.7 Botschaften S. 29](#)) spezifiziert.

#### 3.3.1 Allgemeines Netzwerkmanagement

Innerhalb dieses Punktes (<**net-mgmt-spec**>) können Angaben u.a. zu Sleep-/Wake-up-Mechanismen gemacht werden. Falls das Netz in verschiedenen Betriebsarten laufen kann, so sind diese (und die Übergangsmechanismen) ebenfalls hier zu beschreiben.

Alle Netz-Signale und Botschaften, die für das Netzwerkmanagement eingesetzt werden, müssen als Netzsignale (<**net-signal**> bzw. <**net-message**>) (vgl. [Topic 3.3.6 Netz-Signale S. 27](#)) spezifiziert werden. Dazu dient insbesondere die Zusammenfassung der Einzelbotschaften zu Gruppen (vgl. [Topic 3.3.7 Botschaften S. 29](#)). Bei der Erstellung von Dokumen-

	Strukturelle Grundlagen SNT	Seite: 27/75
	Kapitel: Netz-Signale	Datum: 2002-02-07 Status: RD

tationen sollten Netzwerkmanagementsignale als eigenständige Tabelle (CALS-Tabelle) zusammengefaßt werden<sup>12</sup>.

Ersatzwertbeschaffung wird nicht in die Netzwerkbeschreibung (z.B. bei *Netz-signalgruppe*) aufgenommen, weil diese in der Regel beim Empfänger des Signals auszuführen ist.

Es gibt Aktivitäten bei *ASAP* und bei *OSEK* im Bezug auf Standardisierung des Netzwerkmanagements. Zunächst wird daher auf eine formale Modellierung verzichtet, und nur die vorgeschlagene Kapitelstruktur vorgesehen.

### 3.3.2 Initialisierung

Die Initialisierung (**<net-init-spec >**) beschreibt Dienste und Protokolle, die dazu dienen einen Netzteilnehmer nach Power-up oder Reset in den kommunikationsfähigen Zustand zu bringen.

### 3.3.3 Fehlerbehandlung

Der Abschnitt **<net-error-handling >** dokumentiert die Mechanismen für die Fehlerbehandlung.

### 3.3.4 Diagnose

Der Abschnitt **<net-diag-spec>** dokumentiert die Mechanismen für die Abwicklung der Diagnose auf dem Netz.

### 3.3.5 Blockübertragungsmodi

Die Blockübertragungsmodi (**<net-block-modes >**) beschreiben applikationsspezifische Protokolle (Initialisierung, Diagnose, u. ä.) für den Transfer von Datenpaketen.

Es wird empfohlen, die für die Blockübertragungsmodi verwendeten Botschaften zu referenzieren. Zum Aufbau dieser Botschaften werden ebenfalls Signale verwendet, die über die Netz-Signal-Klasse (**< net-signal-class>**) besonders gekennzeichnet sind.

Es gibt Aktivitäten bei *ISO TC22 SC3 WG1* im Bezug auf Standardisierung der Blockübertragungsmodi. Zunächst wird daher auf eine formale Modellierung verzichtet, und nur die vorgeschlagene Kapitelstruktur vorgesehen.

### 3.3.6 Netz-Signale

**<net-signal-spec >**<sup>13</sup> (s. [Abbildung 11 Netz-Signale S. 28](#)) spezifiziert die auf dem Netz übertragenen Signale.

Die Netz-Signale sind verwandt mit Variablen in einer Steuergerätesoftware. Aus diesem Grund werden optional Teile des Software Datenlexikons (**< sw-units>** und **<sw-compu-methods>**) in die DTD aufgenommen.

<sup>12</sup> eine spätere Erweiterung der DTD könnte die strukturellen Grundlagen für eine Automaitisierung legen.

<sup>13</sup> Zur Abgrenzung gegen signal in MSRDOC wird hier von net-signal gesprochen.

Die Netz-Signal-Gruppen **<net-signal-group >**s dient zur Zusammenfassung von Signalen, die exakt gleiche Eigenschaften haben. Durch die Zusammenfassung in einer Gruppe ist eine redundante Beschreibung nicht mehr notwendig<sup>14 15</sup>.

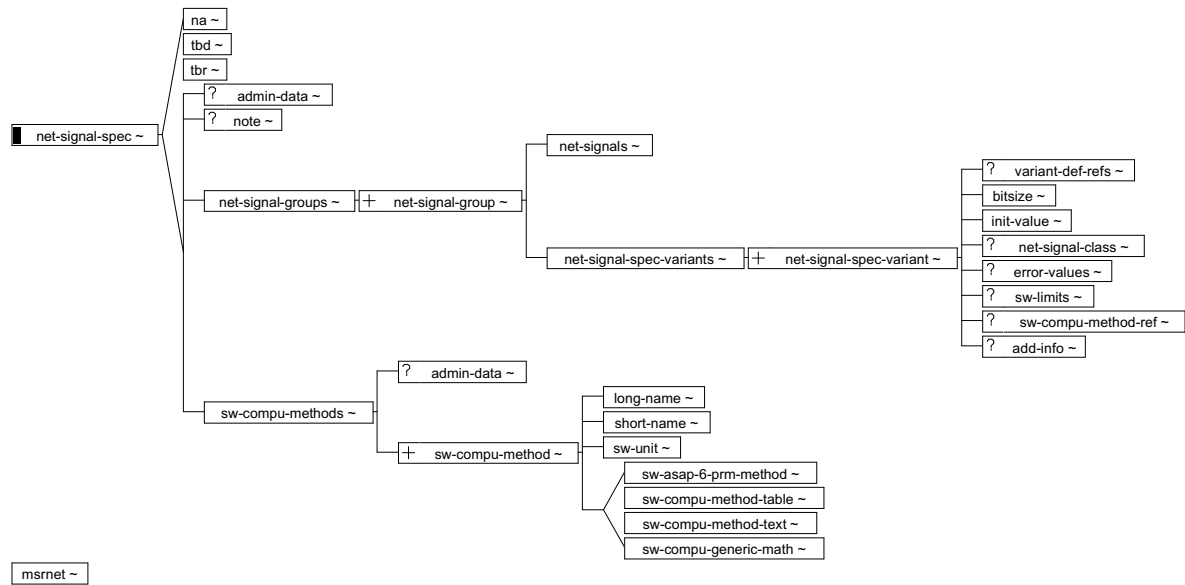


Abbildung 11: Netz-Signale


Die Signal-Eigenschaften (**<net-signal-spec-variant >**) umfassen die folgenden Teile<sup>16</sup>:

- <bitsize>** Länge des Signals in Bit
- <init-value>** Optionaler Initialisierungswert - Dieser Wert wird gesendet, wenn das zugehörige physikalische Signal noch nicht erfaßt worden ist.
- <net-signal-class>** Die optionale Netz-Signal-Klasse (**<net-signal-class>**) kann "Applikationssignal" oder "Netzwerkmanagementsignal" sein.
- <error-values >** Dieser Wert wird gesendet, wenn das zugehörige physikalische Signal aufgrund einer Fehlersituation (z.B. Sensorabfall) nicht zur Verfügung steht. Es können mehrere **<error-value>**s angegeben werden, um verschiedene Fehlersituationen zu unterscheiden. Die Angabe ist optional.
- <sw-limits>** Optionale Angabe des Wertebereich des Signales. Die Angabe kann sowohl in netzinterner (**<coded>**) als auch physikalischer (**<phys>**) Darstellung erfolgen.

<sup>14</sup> Die Struktur ist ähnlich aufgebaut wie signal-group in MSRDOC.DTD

<sup>15</sup> Dies ist ein generelles Prinzip bei MSR

<sup>16</sup> Einige dieser Angaben sind optional, um z.B. auch Block-mode-Signale mit Name und Größe anlegen zu können.

	Strukturelle Grundlagen SNT	Seite: 29/75
	Kapitel: Botschaften	Datum: 2002-02-07 Status: RD

**<sw-compu-method-ref >** Die Umrechnung dokumentiert, wie die netzinterne Darstellung (codierte, bzw. interne Werte) in die zugehörigen physikalischen Werte (externer Werte) umgerechnet wird. Die wird über eine Referenz auf die **<sw-compu-methods >** dokumentiert. Siehe auch: *[Externes Dokument: Strukturelle Grundlagen, Bereich Software / Status: 1.1.0 / Datum: 16.12.97 / Herausgeber: MSR AG-MEDOC / URL: / relevante Stelle: ]*

Dies ist eine optionale Angabe.

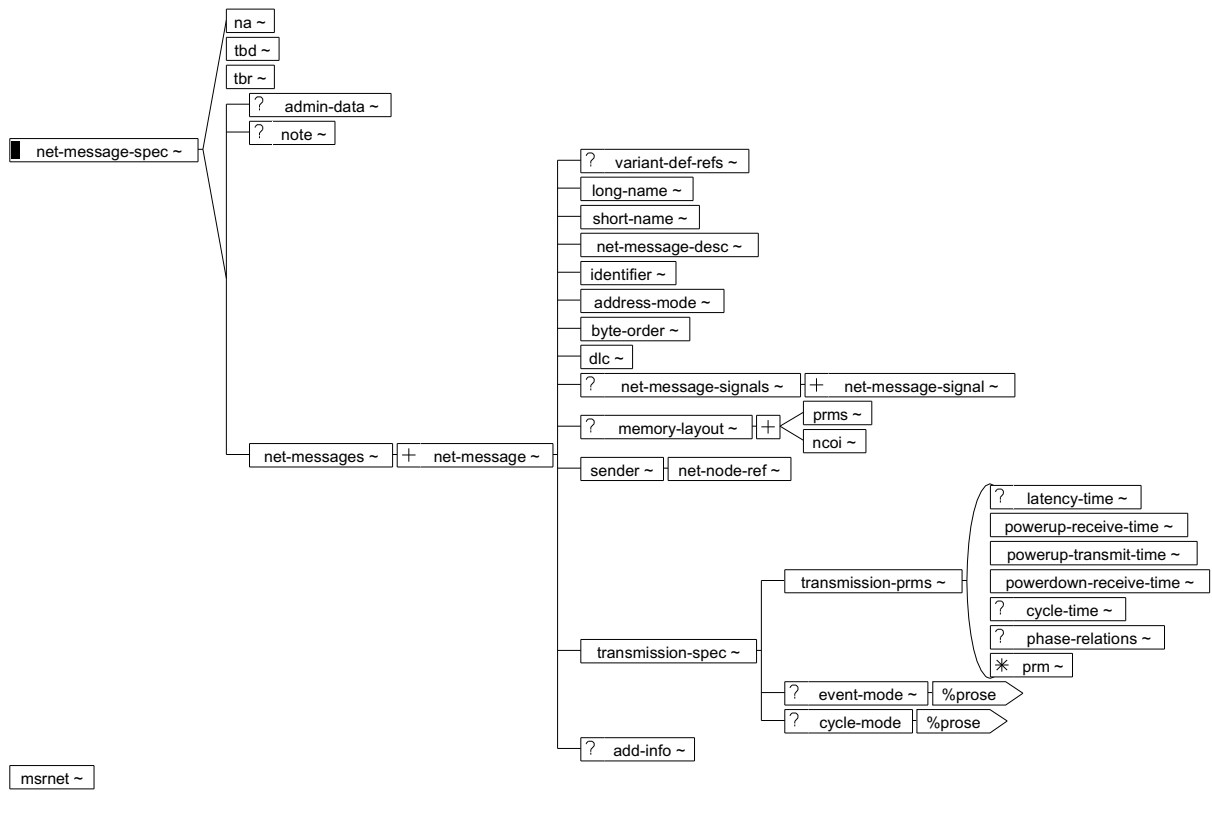
Eine Empfängerliste ist bei der Definition der Netzsignale nicht vorgesehen, da eine solche über die Angaben zu den Botschaften (**<receivers >** innerhalb **<net-message-signals >** innerhalb **<net-message-spec >**) generiert werden kann.

### 3.3.7

## Botschaften

Die Signale auf dem Netz werden in Botschaften zusammengefaßt übertragen. In **<net-message-spec >** (s. [Abbildung 12 Struktur der Botschaften S. 30](#)) können diese Botschaften spezifiziert werden.

Netz Botschaften können in **<net-message-set >**s zusammengefaßt werden. Die Zusammenfassung ist **nicht** für strukturell gleiche Botschaften gedacht. Sie dient vielmehr zur dokumentarischen Zusammenfassung.



**Abbildung 12: Struktur der Botschaften**


Jede Botschaft **<net-message >** wird variantenabhängig spezifiziert und identifiziert durch **< long-name>**, **<short-name>**. Die Beschreibung umfasst:

**<net-message-desc >** Hier hat man die Möglichkeit, die Botschaft in Prosa zu beschreiben.

**<identifier >** Dies ist ein Hex-Wert, über den die Botschaft im Netzwerk identifiziert wird.

Für die Bestimmung der Identifier einer konkreten **<net-message >** (für einen Netz-knoten) werden zwei Mechanismen angeboten:

- Eine diskrete Angabe der Identifier **<net-message-identifiers >**.
- eine Berechnung des Identifiers (**<calc-net-message-identifiers >**) als Summe einer basis und **<Identifier-base-address >** und eines Offset, der im Knoten angegeben wird. Der **< msg-identifier-offset>** wird über beim jeweiligen **<sender >** angegeben.

	Strukturelle Grundlagen SNT	Seite: 31/75
	Kapitel: Botschaften	Datum: 2002-02-07 Status: RD

Dies dient zur Beschreibung von strukturell gleichen (abstrakten) Botschaften. Solche Botschaften können erst dann auf dem Bus übertragen werden wenn sie eine eindeutige Zuordnung zu einem Identifier erhalten.<sup>17</sup>

Es besteht eine 1 zu n - Beziehung zwischen **<identifier >** und **<sender>**, um Varianten beschreiben zu können (**<variant-def-ref>**).

**<address-mode>** Der Address-Mode kann "STD" (Standard) oder "XTD" (Extended) sein.

Eine bessere Bezeichnung ist "CAN-Datenübertragungsformat". Dieses ist eine sehr globale Festlegung für einen Bus und kann "Standard-Format" oder "Extended-Format" sein. Hierzu existieren Spezifikationen der Fa. Bosch. Im Fall von "Extended-Format" kann je Steuergerät/CAN-Controller festgelegt werden, ob es/er "aktiv" oder "passiv" ist, d.h. ob es/er das Extended-Format kommunizieren kann oder nicht.<sup>18</sup>

**<byte-order>** bezeichnet das Byte-Ordering in der Botschaft. Mögliche Werte sind: *motorola*, *motorola-vorwaerts*, *motorola-rueckwaerts* oder *intel*

**<dlc>** Die Angabe DLC (Data Length Code) (siehe [Definition DLC S. 53](#)) wird in Byte angegeben und gibt an, wie lang die Botschaft ist. Eine Botschaft kann Lücken haben. Daher kann die Länge nicht aus der Summe der Signal-Längen ermittelt werden.

**<net-message-signals>** Die Signale einer Botschaft werden in **<net-message-signals >** aufgelistet als (**<net-message-signal>**). Dies ist eine optionale Angabe. **<net-message-signal>** können entweder direkte **<net-signal>**s sein und mit einem **<offset>** versehen werden.


Es kann aber auch gemultiplexte Signale geben. Diese werden als **< multiplex-signal-set>** spezifiziert, welches einen **< multiplexor>** spezifiziert mit **<offset>**<sup>19</sup> und Länge **<bitsize>** besitzt. In der **<multiplex-signal-list >** wird in **<multiplex-entry >** jedem Wert des Multiplexors (**<multiplexor-value>**) wiederum eine Signal-Liste (**<net-message-signals>**) zugeordnet. Damit können beliebig verschachtelte Multiplexsignale aufgebaut werden<sup>20</sup>.

<sup>17</sup> Solche Botschaften sind z.B. Kommandobotschaften mit denen eine bestimmte Funktion innerhalb eines Steuergerätes von verschiedenen Knoten angestoßen wird. Weitere Beispiele sind Netzwerkmanagement- und Diagnosebotschaften.

<sup>18</sup> Es muß noch geklärt werden, wo diese Festlegung stattfindet.

<sup>19</sup> Die offsets beziehen sich grundsätzlich auf den Botschaftsanfang und werden addiert (beginnen also bei 0)

<sup>20</sup> Diese Hierarchie ist auf allen Ebenen variantenabhängig möglich. Variantenabhängigkeiten sollten jedoch nur auf oberster Ebene eingeführt werden.

	Strukturelle Grundlagen SNT	Seite: 32/75
	Kapitel: Botschaften	Datum: 2002-02-07 Status: RD

Bei jedem Signal einer Botschaft (direkt oder gemultiplext) sind auch die zugehörigen Empfänger **<receivers>** zu dokumentieren, da diese durchaus von der Botschaft abhängig sein können.

**<net-message-layout>** Die Angabe **<net-message-layout >** ist als optionale, redundante Ergänzung zu der Signalliste (**< net-message-signals >**) vorgesehen. Dort können vorprozessierte Informationen abgelegt werden, die nicht vom *SGML-Formatierer* aus den **<net-message-signals>** abgeleitet werden können. Gültig sind stets die Angaben unter **<net-message-signals>**.

**<sender>** Bezeichnet den Knoten, welcher das Signal sendet.

**<transmission-spec>** beschreibt die Übertragungsverfahren der Botschaft. Es gelten folgende Parameter

**<powerup-receive-time>** [Definition Empfängerbereitschaft nach Power up S. 53](#)  
Empfängerbereitschaft nach Power up

Übertragungsparameter einer Botschaft, der den Zeitpunkt nach *Power up* beschreibt, ab wann die Botschaft empfangen werden kann.


**<powerup-transmit-time>** [Definition Sendebereitschaft nach Power up S. 54](#)  
Empfangsbereitschaft nach Power down

Übertragungsparameter einer Botschaft, der den Zeitpunkt nach *Power up* beschreibt, ab wann die Botschaft gesendet werden kann.

**<powerdown-receive-time>** [Definition Empfangsbereitschaft nach Power down S. 53](#)  
Empfangsbereitschaft nach Power down

Übertragungsparameter einer Botschaft, beschreibt die Zeit nach einer *Power down Meldung*, in der zwar nicht mehr gesendet, aber noch auf eintreffende Botschaften reagiert werden kann. Damit wird vermieden, daß eine Komponente in den *sleep mode* geht, obwohl gerade eine Botschaft an sie unterwegs ist.



	Strukturelle Grundlagen SNT	Seite: 33/75
	Kapitel: Botschaften	Datum: 2002-02-07 Status: RD

### <latency-time>

[Definition Latenzzeit S. 53](#)  
Latenzzeit

Optionaler Übertragungsparameter einer Botschaft, der die Übertragungszeit auf dem Netz beschreibt. Die Latenzzeit ist die Zeit, um der Sendevorgang einer Botschaft durch höherpriorie Botschaften verzögert werden kann. Technisch gesprochen ist die Latenzzeit die Zeitspanne, die zwischen TransmitRequest-Bit setzen und AckMeldung erhalten liegt. An der Botschaft kann also die maximale Latenzzeit für diese Botschaft spezifiziert werden<sup>21</sup>.

### <cycle-time>

[Definition Zykluszeit S. 54](#)  
Zykluszeit (auch Zyklustoleranzzeit), optionaler Übertragungsparameter einer Botschaft, der die Toleranzzeit einer Botschaft bei zyklischer Steuerung (zyklisches Versenden der Botschaft) beschreibt<sup>22</sup>.

### <phase-relations>

[Definition Phasenbeziehung zu anderen Botschaften S. 54](#)  
Phasenbeziehung zu anderen Botschaften

Übertragungsparameter einer Botschaft, der beschreibt, ob eine Phasenbeziehung zu anderen Botschaften vorliegt.

<sup>21</sup> nur bei zyklischer Steuerug relevant, daher optional

<sup>22</sup> nur bei zyklischer Steuerug relevant, daher optional

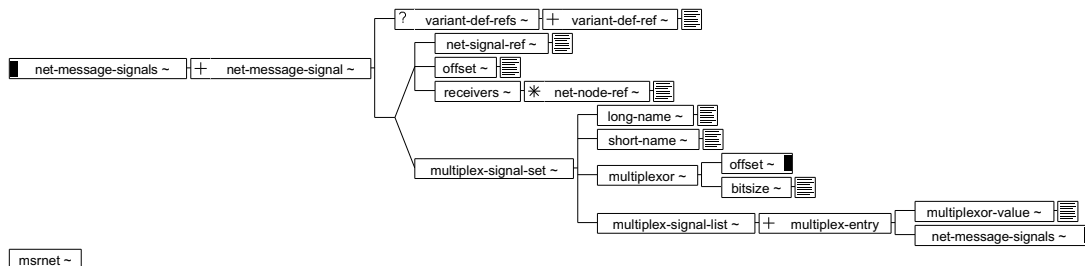


Abbildung 13: Aufbau der signale in einer Botschaft

Die Konsistenz der eingegebenen Daten ist durch eine semantische Prüfung zu gewährleisten.

## 3.4 Varianten

Variantenabhängigkeiten existieren bei:

- Knoten (< **net-node**>)
- Netinterface< **interface-circuit**>
- Netzsegmente (< **segment**>)
- Netz-Signal < **net-signal**>
- Netz-Signal-Eigenschaften <**net-signal-group**>
- Botschaft (< **net-message**>) wird dann verwendet, wenn die Botschaft variantenabhängig vorhanden ist.
- Signaleintrag in Botschaft (<**net-message-signal**>) wird dann verwendet, wenn das Signal variantenabhängig vorhanden ist.
- Speicherlayout (Aufgabe für AG-DTD %info variantenabhängig könnte man mit <**xref**> auf Variantendefinition) realisieren.
- Übertragungsparameter (<**transmission-spec**>) <sup>23</sup>
- Für die Bestimmung der Identifier einer einer konkreten < **net-message**> werden zwei Mechanismen angeboten:
  - Eine diskrete Angabe der Identifier <**net-message-identifiers**>.
  - eine Berechnung des Identifiers (<**calc-net-message-identifiers**>) als Summe einer basis und <**identifier-base-adress**> und eines Offset, der im Knoten angegeben wird. Der < **msg-identifier-offset**> wird über beim jeweiligen <**sender**> angegeben. Dieses Verfahren wird vorzugsweise für Netzwerkmanagement und Diagnose verwendet.

TBD

<sup>23</sup> in 11.0 noch nicht realisiert

## 4 Übersicht Changes

[snt] Strukturelle Grundlagen Netzwerke

### 4.1 Übersicht zu Zweite Version mit neuem Anwendungsprofil

geplant für ???

**Tabelle 3: Lösungen in Zweite Version mit neuem Anwendungsprofil**

Name	Art	Thema	Status	Prio	MT	Aufgetreten in	S.
cmt-text		sw-compu-method-text mit Beschreibung	angenommen			Sitzung 12.12.96	-> S. 44
resp.obj		Zuweisung von Verantwortlichkeiten zu Objekten	angenommen			Sitzung 12.12.96	-> S. 49

#### 4.1.1 Release notes

##### 4.1.1.1 [cmt-text] sw-compu-method-text mit Beschreibung

<sw-compu-method-text> can now be documented in detail by using <desc> within <sw-compu-method-value-pair >.

Migration No migration required

##### 4.1.1.2 [resp.obj] Zuweisung von Verantwortlichkeiten zu Objekten

<admin-data> is now allowed on < net-port-variant>

### 4.2 Übersicht zu Erste Version mit Bugfixes

geplant für 06.02.98

**Tabelle 4: Requests zu Erste Version mit Bugfixes**

Name	Art	Thema	Status	Prio	MT	geplant für	S.
trans-mode	enh	Transmission-Modes ausmodellieren	<b>Offen</b> H. Riegraf macht einen Vorschlag.			Erste Version mit Bugfixes	-> S. 43
docrefopt	enh	Dokumentübergreifende Verweise optional	<b>angenommen</b>			Erste Version mit Bugfixes	-> S. 46

**Tabelle 5: Lösungen in Erste Version mit Bugfixes**

Name	Art	Thema	Status	Prio	MT	Aufgetreten in	S.
trans-mode	enh	Transmission-Modes ausmodellieren	<b>Offen</b> H. Riegraf macht einen Vorschlag.			Erste Version mit Bugfixes	-> S. 43
basetype		sw-base-type in net-signal	<b>angenommen</b>			Sitzung 12.12.96	-> S. 44
docrefopt	enh	Dokumentübergreifende Verweise optional	<b>angenommen</b>			Erste Version mit Bugfixes	-> S. 46
init-opt	enh	init-value optional	<b>angenommen</b>			Erste Version	-> S. 46
msg-set	enh	Botschaften in einzelne Kapitel	<b>angenommen</b>	A 1		Erste Version	-> S. 47
msgid-mask	enh	Gleiche Botschaft aus verschiedenen Sendern	<b>angenommen</b>			Sitzung 12.12.96	-> S. 47

## 4.2.1 Release notes

### 4.2.1.1 enh: [trans-mode] Transmission-Modes ausmodellieren

<event-mode> und <cycle-mode> wird entfernt bis formae Modelle vorliegen.

#### 4.2.1.2 [basetype] sw-base-type in net-signal

<net-signal-spec-variant> now has < sw-base-type>.

#### 4.2.1.3 enh: [docrefopt] Dokumentübergreifende Verweise optional

Die potentiell dokumentübergreifende Verweise werden optionalisiert und mit einem Label versehen. Damit kann *MSRNET.DTD* autonom benutzt werden.

#### 4.2.1.4 enh: [init-opt] init-value optional

<init-value> ist nun optional in < net-signal-spec-variant>

#### 4.2.1.5 enh: [msg-set] Botschaften in einzelne Kapitel

<memory-layout> heißt nun <net-message-layout > umbenannt.

Netz Botschaften können nun in <net-message-set>s zusammengefaßt werden. Die Zusammenfassung ist für inhaltlich gleiche Botschaften gedacht. Sie dient zur dokumentarischen Zusammenfassung.

Wenn eine solche Instanz in eine nicht normalisierte Datenbank eingelesen wird (d.h. die <short-name> der < net-message-set> werden den <net-message> zugeordnet), dient beim Wiederaufbau einer MSRNET-Instanz der <short-name > als Gruppenwechselkriterium.

#### 4.2.1.6 enh: [msgid-mask] Gleiche Botschaft aus verschiedenen Sendern

### 4.3 Übersicht zu Erste Version

geplant für 18.11.97

Tabelle 6: Requests zu Erste Version

Name	Art	Thema	Status	Prio	MT	geplant für	S.
non-can		Ausdehnung auf nicht CAN-Netze	Offen				-> S. 42
sw-var-abgl		Abgleich zwischen MSRSW-Variablen und MSRNET Signalen	Offen				-> S. 43

**Tabelle 6 (Forts.): Requests zu Erste Version**

Name	Art	Thema	Status	Prio	MT	geplant für	S.
init-opt	enh	init-value optional	angenommen			Erste Version mit Bugfixes	-> S. 46
msg-set	enh	Botschaften in einzelne Kapitel	angenommen	A 1		Erste Version mit Bugfixes	-> S. 47

**Tabelle 7: Lösungen in Erste Version**

Name	Art	Thema	Status	Prio	MT	Aufgetreten in	S.
ctrl-mode	enh	Wo wird der Mode der Controller festgelegt	angenommen			Sitzung 12.12.96	-> S. 45
var-trans	bug	Varianten in transmission-spec	angenommen			Sitzung 12.12.96	-> S. 50

## 4.3.1 Release notes

### 4.3.1.1 enh: [ctrl-mode] Wo wird der Mode der Controller festgelegt

<adress-mode> is now renamed to < identifier-type> which can take the values *standard* bzw. *extended*.

<node-variant> now has <protocol-conformance > to capture the level of conformance the node provides.

### 4.3.1.2 bug: [var-trans] Varianten in transmission-spec

<transmission-spec> is no longer variant dependant.

## 4.4 Übersicht zu Einführung der Net-Ports

geplant für 12.6.96

**Tabelle 8: Requests zu Einführung der Net-Ports**

Name	Art	Thema	Status	Prio	MT	geplant für	S.
Netport-Node	enh	Behandlung von Knoten mit mehreren Netports zu unflexibel	angenommen			Sitzung 22.8.96	-> S. 48

## 4.5 Übersicht zu Sitzung 22.8.96

geplant für 22.8.96

**Tabelle 9: Lösungen in Sitzung 22.8.96**

Name	Art	Thema	Status	Prio	MT	Aufgetreten in	S.
Netport-Node	enh	Behandlung von Knoten mit mehreren Netports zu unflexibel	<b>angenommen</b>			Einführung der Net-Ports	-> S. 48

## 4.6 Übersicht zu Sitzung 12.12.96

geplant für 12.12.96

**Tabelle 10: Requests zu Sitzung 12.12.96**

Name	Art	Thema	Status	Prio	MT	geplant für	S.
gateway		Wie wird Gateway-Funktionalität behandelt	<b>Offen</b>				-> S. 41
multi-arch	enh	Unterstützung von Multi-Architektur-Netzen	<b>Offen</b> BMW, V-W, Porsche diskutieren weiter				-> S. 41
net-class	enh	Netzwerk - Protokollklassen	<b>Offen</b>				-> S. 42
supersignal		Supersignal	<b>Offen</b>				-> S. 42
basetype		sw-base-type in net-signal	<b>angenommen</b>			Erste Version mit Bugfixes	-> S. 44
cmt-text		sw-compu-method-text mit Beschreibung	<b>angenommen</b>			Zweite Version mit neuem Anwendungsprofil	-> S. 44
ctrl-mode	enh	Wo wird der Mode der Controller festgelegt	<b>angenommen</b>			Erste Version	-> S. 45

**Tabelle 10 (Forts.): Requests zu Sitzung 12.12.96**

Name	Art	Thema	Status	Prio	MT	geplant für	S.
msgid-mask	enh	Gleiche Botschaft aus verschiedenen Sendern	<b>angenommen</b>			Erste Version mit Bugfixes	-> S. 47
resp.obj		Zuweisung von Verantwortlichkeiten zu Objekten	<b>angenommen</b>			Zweite Version mit neuem Anwendungsprofil	-> S. 49
seg.na	enh	Segmentierung wird nicht immer eingegeben	<b>angenommen</b>	A1			-> S. 49
var-trans	bug	Varianten in transmission-spec	<b>angenommen</b>			Erste Version	-> S. 50
Ink-sw		Wie werden Signale zu SW-Variablen zugeordnet	<b>abgelehnt</b>				-> S. 50



## 5 Changes

### 5.1 [gateway] Wie wird Gateway-Funktionalität behandelt

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
	Rauleder	Offen			[11.0] S. 39	

Sache Wie werden Gateway - Funktionen beschrieben? Wie kommen Links zu *MSRSW.DTD* Instanzen. Dabei ist nur die Signal/Signal-Zuordnung zu betrachten?

Begründung Hintergrund der Frage ist der unterschiedliche Grad der Formalisierung von Signalen im Spannungsfeld zwischen *MSRSW.DTD* *MSRNET.DTD* *MSW-DOC.DTD*

#### Beschlossene Lösung

muß noch erarbeitet werden

### 5.2 [multi-arch] Unterstützung von Multi-Architektur-Netzen

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
enh	BMW H. Kramer	Offen BMW, VW, Porsche diskutieren weiter			[11.0] S. 39	

Sache Es muß eine Möglichkeit geben, Netzwerke zu beschreiben, die mit unterschiedlichen Protokollen und Architekturen arbeiten. Dabei ist zu betrachten

Gateway Das Gesamtnetz besteht aus mehreren Beschreibungen mit getrennten Signaldefinitionen. Die Teilnetze wissen nichts voneinander.

Repeater Das Gesamtnetz hat eine konsistente Beschreibung und eine gemeinsame Signaldefinition.

Begründung Die Szenarien treten in der Praxis auf, in erster Linie aber Gateways

### Lösungsansatz 1

Es werden mehrere Netz-Spezifikationen eingeführt, wobei eine gemeinsame und Subnetz-Komponenten auftreten. Die Subnetz-Komponenten können auf das gemeinsame Netz verweisen. Ohne Subnetze ist das gemeinsame Netz das gesamte Netz. Es ergibt sich also folgende Struktur.

```

common-network
  net-signal-spec
  net-message-spec
sub-net*
  net-signal-spec
  net-signal
  net-signal-ref
  net-message-spec
  net-message
  net-message-ref
  
```

Zusätzlich ist ein Modell für Gateways zu finden.

## 5.3 [net-class] Netzwerk - Protokollklassen

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
enh	BMW H. Kramer	Offen			<a href="#">[11.0] S. 39</a>	

Sache Es muß die Möglichkeit geben, den Protokolltyp eines Netzes zu spezifizieren.

Begründung Es gibt verschiedenartigen Netze

## 5.4 [non-can] Ausdehnung auf nicht CAN-Netze

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
	AG-Net 18.11.97	Offen			<a href="#">[1.1.0] S. 37</a>	

Sache Die Ausweitung auf nicht-CAN-Netze muß eingebaut werden.

Begründung Es gibt auch nicht-CAN

## 5.5 [supersignal] Supersignal

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
	BMW H. Kramer	Offen			[11.0] S. 39	

Sache Es muß die Möglichkeit geben, verschiedenartige Signale zusammenzufassen.

Begründung Verbesserung der Dokumentation.

### Lösungsansatz 1

Lösung analog zu `<port-groups>` in MSRDOC

### Lösungsansatz 2

Signale können über ein Cluster zusammengefasst werden. Dies ist ein Element mit mehreren Zeigern auf `<net-signal>`.

## 5.6 [sw-var-abgl ] Abgleich zwischen MSRSW-Variablen und MSRNET Signalen

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
	AG Net 18.11.97	Offen			[1.1.0] S. 37	

Sache In künftigen Versionen muß ein Abgleich zwischen `<sw-variable >` und `<net-signal-spec-variant>`

Begründung Diese Daten stehen in enger Beziehung zueinander.

Siehe auch **sonst:** [Kapitel 5.8 \[basetype\] sw-base-type in net-signal S. 44](#)

## 5.7 [trans-mode] Transmission-Modes ausmodellieren

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
enh	Riegraf, Aufdermauer	<b>Offen</b> H. Riegraf macht einen Vorschlag.			[1.1.1] S. 35	[1.1.1] S. 35

Sache Die transmission-Modes **<event-mode>** bzw. **< cycle-mode>** sollten ausmodelliert werden

Begründung Damit kann eine formale Behandlung in Tools erfolgen

### Beschlossene Lösung

Zunächst die informalen **<event-mode>** und **< cycle-mode>** entfernen. Wenn ein formaler Vorschlag vorliegt, wird er eingebaut

### Release notes

1.1.1enh: [trans-mode] Transmission-Modes ausmodellieren S. 36

## 5.8 [basetype] sw-base-type in net-signal

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
	BMW H. Krammer	<b>angenommen</b>			[11.0] S. 39	[1.1.1] S. 35

Sache Es sollte die Möglichkeit geben, einen Basistyp für ein Signal anzugeben

Begründung Aus der Umrechnungsformel allein kann man nicht erkennen, ob es ein vorzeichenbehafteter Wert sein kann.


### Beschlossene Lösung

Es wird **<sw-base-type>** in **<net-signal-spec-variant >** eingebaut.

In künftigen Versionen muß ein Abgleich zwischen **<sw-variable >** und **<net-signal-spec-variant>**

### Release notes

1.1.1[basetype] sw-base-type in net-signal S. 36

	Strukturelle Grundlagen SNT	Seite: 45/75
	Kapitel: [ctrl-mode ] Wo wird der Mode der Controller festgelegt	Datum: 2002-02-07 Status: RD

## 5.9 [cmt-text] sw-compu-method-text mit Beschreibung

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
	BMW H. Kramer	angenommen			[11.0] S. 39	[2.1.0] S. 35

Sache **<sw-compu-method-text>** sollte ein weiteres Element **<cmt-desc>** haben, in welches weitere Informationen zu einem bestimmten Wert geschrieben werden kann.

Begründung Es kann sein daß **<cmt-text>** ein formaler Text ist, welcher in der Dokumentation detaillierter beschrieben werden soll.

### Beschlossene Lösung

Es wird ein Element **<desc>** innerhalb **<sw-compu-method-value-pair>** eingeführt.

### Release notes

[2.1.0\[cmt-text\] sw-compu-method-text mit Beschreibung S. 35](#)

## 5.10 [ctrl-mode ] Wo wird der Mode der Controller festgelegt

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
enh	Riegraf / Weichel	angenommen			[11.0] S. 39	[1.1.0] S. 37

Sache Aussage von [Mitarbeiter Dipl.-Ing. T. Riegraf S. 8](#) zu **<identifizier-type>**: Eine bessere Bezeichnung ist "CAN-Datenübertragungsformat". Dieses ist eine sehr globale Festlegung für einen Bus und kann "Standard-Format" oder "Extended-Format" sein. Hierzu existieren Spezifikationen der Fa. Bosch. Im Fall von "Extended-Format" kann je Steuergerät/CAN-Controller festgelegt werden, ob es/er "aktiv" oder "passiv" ist, d.h. ob es/er das Extended-Format kommunizieren kann oder nicht. Es muß noch geklärt werden, wo diese Festlegung stattfindet.

Begründung Sonst haben wir eine Lücke

### Beschlossene Lösung

- <adress-mode>** wird zu **<identifizier-type >** umbenannt. Dieser kann die Werte *standard* bzw. *extended* annehmen.

- **<node-variant>** bekommt ein neues Element **<protocol-conformance>** und kan die folgenden Werte annehmen

CAN-2.0A-active [Definition CAN Conformance 2.0A S. 52](#)CAN Conformance 2.0A

CAN-2.0B-active [Definition CAN Conformance 2.0B S. 52](#)CAN Conformance 2.0B

CAN-2.0B-passiv [Definition CAN Conformance 2.0B passiv S. 52](#)CAN Conformance 2.0B passiv

NA Das beschriebene Netz kenn keine Konformitätsklassen. Der Wert ist daher nicht ausfüllbar.

#### Release notes

[1.1.0enh: \[ctrl-mode \] Wo wird der Mode der Controller festgelegt S. 38](#)

## 5.11

### [docrefopt] Dokumentübergreifende Verweise optional

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
enh	Klein	angenommen			<a href="#">[1.1.1] S. 35</a>	<a href="#">[1.1.1] S. 35</a>

Sache Die Dokumentübergreifenden Verweise sollten optional sein.

Begründung Vereinfacht die Handhabung in einfachen Autorenumgebungen. Beschluß in der AG-DTD

#### Beschlossene Lösung

Die potentiell dokumentübergreifende Verweise werden optionalisiert und mit einem Label versehen. Damit kann *MSRNET.DTD* autonom benutzt werden.


#### Release notes

[1.1.1 enh: \[docrefopt\] Dokumentübergreifende Verweise optional S. 37](#)

## 5.12

### [init-opt] init-value optional

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
enh	riegraf	angenommen			<a href="#">[1.1.0] S. 37</a>	<a href="#">[1.1.1] S. 35</a>

	Strukturelle Grundlagen SNT	Seite: 47/75
	Kapitel: [msgid-mask] Gleiche Botschaft aus verschiedenen Sendern	Datum: 2002-02-07 Status: RD

Sache **<init-value>** soll optional sein.

Begründung Die Angabe kann nicht immer gemacht werden, insbesondere bei Blockmoden.

#### Beschlossene Lösung

**<init-value>** ist nun optional in **< net-signal-spec-variant>**

#### Release notes

1.1.1 enh: [\[init-opt\] init-value optional S. 37](#)

## 5.13 [msgid-mask] Botschaften in einzelne Kapitel

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
enh	Kramer	angenommen	A 1		<a href="#">[1.1.0] S. 37</a>	<a href="#">[1.1.1] S. 35</a>

Sache Es sollte eine Möglichkeit geben, Netzbotschaften so zu kategorisieren, daß sie bei gedruckter Dokumentation auf einzelne Kapitel verteilt werden können.

Begründung Damit wird die Dokumentation übersichtlicher

#### Beschlossene Lösung


Es wird auch für Botschaften ein Gruppenkonzept eingeführt **< net-message-set>**. Es wurde nicht "group" verwendet, da "group" konzeptuell gleichartige Dinge zusammenfaßt. Nur bei der *MSRSYS.DTD* gibt es für Ports die Möglichkeit "group" zur Inhaltlichen und logischen Zusammenfassung zu verwenden.

#### Release notes

1.1.1enh: [\[msgid-set\] Botschaften in einzelne Kapitel S. 37](#)

## 5.14 [msgid-mask] Gleiche Botschaft aus verschiedenen Sendern

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
enh	BMW H. Kramer	angenommen			<a href="#">[11.0] S. 39</a>	<a href="#">[1.1.1] S. 35</a>

	Strukturelle Grundlagen SNT	Seite: 48/75
	Kapitel: [ Netport-Node] Behandlung von Knoten mit mehreren Netports zu unflexibel	Datum: 2002-02-07 Status: RD

**Sache** Es muß möglich sein, einer Botschaft mehrere Sender zuzuweisen. Diese unterscheiden sich hinsichtlich Variante und Identifier.

**Begründung** Es kann sein, daß die gleiche Botschaft von mehreren Sendern und unterschiedliche Identifier verschickt wird. Die Empfänger reagieren auf einen maskierten Identifier. Beispiel ist Fensterheber, wo jeder Taster die gleiche Botschaft abdrückt.

Weiterhin ist Varianten handhabung zu unterstützen.

### Lösungsansatz 1

Für die Bestimmung der Identifier einer einer konkreten **< net-message>** werden zwei Mechanismen angeboten:

- Eine diskrete Angabe der Identifier **<net-message-identifiers >**.
- eine Berechnung des Identifiers (**<calc-net-message-identifiers >**) als Summe einer basis und **<Identifier-base-adress >** und eines Offset, der im Knoten angegeben wird. Der **< msg-identifier-offset>** wird über beim jeweiligen **<sender >** angegeben. Dieses Verfahren wird vorzugsweise für Netzwerkmanagement und Diagnose verwendet.

### Beschlossene Lösung

Es wird auch für Botschaften ein Gruppenkonzept eingeführt **< net-message-group>**

### Release notes

1.1.1enh: [\[msgid-mask\] Gleiche Botschaft aus verschiedenen Sendern S. 37](#)

## 5.15

### [ Netport-Node] Behandlung von Knoten mit mehreren Netports zu unflexibel

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
enh	??	angenommen			<a href="#">[10.01] S. 38</a>	<a href="#">[10.02] S. 38</a>


**Sache** Zur Zeit ist es so, daß jeder Net-Port als eigener Knoten in der Segmentierung auftritt.

**Begründung** Das ist zu unflexibel. Der Zusammenhang zwischen den Netports ist im Netz-Dokument nicht sichtbar.

### Lösungsansatz 1

Man könnte im Knoten auf die Erwähnung des Net-ports verzichten, dafür bei der Referenzierung des Knotens in der Segmentierung den Netport mit angeben.



	Strukturelle Grundlagen SNT	Seite: 49/75
	Kapitel: [seg.na] Segmentierung wird nicht immer eingegeben	Datum: 2002-02-07 Status: RD

### Beschlossene Lösung

Net-ports wurde innerhalb von Knoten eingerichtet (s. [Topic 3.2.2.2 Beschreibung der Netzknoten S. 21](#)). Die Segmentierung verweist auf die `< net-node-port >`s.

### Release notes

## 5.16 [resp.obj] Zuweisung von Verantwortlichkeiten zu Objekten

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
	BMW H. Kramer	angenommen			[11.0] S. 39	[ 2.1.0] S. 35

Sache Es sollte möglich sein, die Verantwortung für bestimmte Teilbäume explizit anzugeben.

Begründung Die beteiligten müssen ihre Partner kennen.

### Lösungsansatz 1

`<Admin-data>` wird weiter verbreitet, in `< company-doc-info >` wird ein `<team-member-ref>` eingeführt.

### Beschlossene Lösung

Bei den der `<Team-Member>`s werden verweise auf Objekte eingeführt, ähnlich wie `<role>`.


`<admin-data>` wird auch bei `<net-node-variant >` eingeführt

### Release notes

[2.1.0\[resp.obj\] Zuweisung von Verantwortlichkeiten zu Objekten S. 35](#)

## 5.17 [seg.na] Segmentierung wird nicht immer eingegeben

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
enh	BMW. H. Kramer	angenommen	A1		[11.0] S. 39	

	Strukturelle Grundlagen SNT	Seite: 50/75
	Kapitel: [Ink-sw ] Wie werden Signale zu SW-Variablen zugeordnet	Datum: 2002-02-07 Status: RD

Sache Segmentierung sollte auch **<na>** erlauben. Im Prinzip sollte man das so weit treiben, daß verschiedene Szenarien abgedeckt werden können, ähnlich wie bei msrsw.

Begründung Es kann sein, daß das Netz abstrakt beschrieben werden soll, also ohne Segmentierung. Das erlaubt die DTD derzeit nicht.

#### Beschlossene Lösung

**<setmentation>** und ihre Zwillinge bekommen **< na>** usw.

## 5.18 [var-trans] Varianten in transmission-spec

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
bug	Weichel	angenommen			[11.0] S. 39	[1.1.0] S. 37

Sache **<transmission-spec>** ist noch nicht variantenabhängig, ob wohl [Topic 3.4 Varianten S. 34](#) das behauptet

Begründung Das ist inkonsistent

#### Beschlossene Lösung

Dokumentation korrigiert in [Topic 3.4 Varianten S. 34](#).

#### Release notes

1.1.0 bug: [\[var-trans\] Varianten in transmission-spec S. 38](#)

## 5.19 [Ink-sw ] Wie werden Signale zu SW-Variablen zugeordnet

Art	Vorgeschlagen von	Status	Prio	Tage	Aufgetreten in	Geplant für
	Rauleder 18.1.97	abgelehnt			[11.0] S. 39	

Sache Wie kommen Links zu *MSRSW.DTD* Instanzen. Dabei ist nur die Signal/Signal-Zuordnung zu betrachten?

Begründung Wir haben keine Möglichkeit, diese Zuordnung durchzuführen.

	Strukturelle Grundlagen SNT Kapitel: [Ink-sw ] Wie werden Signale zu SW-Variablen zugeordnet	Seite: 51/75 Datum: 2002-02-07 Status: RD
---	--	---


### Lösungsansatz 1

Parallel zu **<locs>** wird **<net-sig-var-assocs >** eingeführt. Dort kann an einer Stelle die Netz-Signal zu Variable-Zuordnung beschrieben werden.

- Pro
- Löst das gegebene Problem
- Con
- Diese Angaben gehören eher zur Software und sollten dort beschrieben werden.

### Beschlossene Lösung

Soll bei *MSRSW.DTD* behandelt werden.

	Strukturelle Grundlagen SNT	Seite: 52/75
	Kapitel: Glossar	Datum: 2002-02-07 Status: RD

## Anh. A Glossar

**Abtastzeitpunkt** Bestimmt als Netzparameter den Zeitpunkt der Bitabtastung.

**Adress-Mode** Der Adress-Mode einer Botschaft kann "Standard" oder "Extended" sein.

**ASAP** Die ASAP-Schnittstellen wurden vom "Arbeitskreis zur Standardisierung von Applikationssystemen (ASAP)" vereinbart. Mitglieder dieses Arbeitskreises sind deutsche Automobilhersteller und Firmen der Zuliefererindustrie.

**Basic-CAN** CAN-Chip; besitzt nur einen Sende- und einen Empfangsspeicher.

**Baudrate**

**Betriebsart** Übertragungsparameter einer Botschaft, der z.B. beschreibt, ob ein Remote-Betrieb vorliegt oder nicht.

**Bit Stuffing** Eine Technik, die in bitorientierten Netzprotokollen zur Resynchronisation genutzt wird. Nach einer festgelegten Anzahl von Bits gleichen Pegels wird durch Einfügen eines zusätzlichen komplementären Bits ein Flankenwechsel hervorgerufen. Das zusätzliche Bit wird vom Empfänger wieder entfernt.

**BRT** Abkürzung für Baudrate Prescaler, definiert die Anzahl BTL-Zyklen.

**BTL-Zyklen** Abkürzung für Bit Timing Logic-Takt; über den Baudrate Prescaler wird aus der Oszillatorfrequenz des Chips ein Abtastraster gebildet. Dieser Hilfstakt (BTL-Takt) dient zur Bestimmung der Zeitsegmente des Bit-Timings.

**Byte-Order** Netzparameter, der die Reihenfolge für das höherwertige und das niederwertige Byte innerhalb eines Wortes festlegt. Da die Prozessor-Hersteller "Intel" und "Motorola" die Byte-Order konträr festgelegt haben, sind die möglichen Werte für diesen Parameter "Intel", "Motorola vorwärts" und "Motorola rückwärts".


**CAN** Abkürzung für Controller Area Network

**CAN High Speed** Netzankopplung und Netzmedium nach [ / Norm: *To be defined* / Status: *DIS* / URL: / relevante Stelle: *ganzes Dokument*], u.a. gekennzeichnet durch eine Datenrate von 125kBit/s bis 1Mbit/s.

**CAN Low Speed** Netzankopplung und Netzmedium nach [ / Norm: *To be defined* / Status: *DIS* / URL: / relevante Stelle: *gesamtes Dokument*], u.a. gekennzeichnet durch eine Datenrate bis 125kBit/s.

**CAN Conformance 2.0A** CAN mit einer Identifierlänge von 11 Bit (Standard).

**CAN Conformance 2.0B** CAN mit einer Identifierlänge von 29 Bit (Extended). Alle CAN-Controller, die CAN 2.0B-Telegramme verarbeiten können, sind auch in der Lage Standard-Frames zu verarbeiten, d.h. sowohl zu senden als auch zu empfangen.

	Strukturelle Grundlagen SNT	Seite: 53/75
	Kapitel: Glossar	Datum: 2002-02-07 Status: RD

**CAN Conformance 2.0B passiv** CAN-Controller, die nur Standard-Frames verarbeiten und sich bei CAN Version 2.0B passiv verhalten. Sie ignorieren die Extended-Telegramme nach Version 2.0B.

**CANdb** CAN-Datenbasis

**CRC** Abkürzung für Cyclic Redundancy Check; enthält den Fehlercode über alle vorangegangenen Stellen. Die CRC-Prüfsumme wird zur Fehlererkennung, nicht zur Fehlerkorrektur verwendet.

**CSMA/CD** Abkürzung für Carrier Sense Multiple Access with Collision Detect

**DLC** Abkürzung für Data Length Code; gibt die Länge der Botschaft in Byte an.

**Empfangsbereitschaft nach Power down** Übertragungsparameter einer Botschaft, beschreibt die Zeit nach einer *Power down Meldung*, in der zwar nicht mehr gesendet, aber noch auf eintreffende Botschaften reagiert werden kann. Damit wird vermieden, daß eine Komponente in den sleep mode geht, obwohl gerade eine Botschaft an sie unterwegs ist.

**Empfängerbereitschaft nach Power up** Übertragungsparameter einer Botschaft, der den Zeitpunkt nach Power up beschreibt, ab wann die Botschaft empfangen werden kann.

**Fehlerwert** Wert eines Signals, wenn ein Fehler erkannt wird und die eigentliche Informationen nicht übertragen werden kann.

**Full-CAN** CAN Chip, besitzt im Gegensatz zum Basic-CAN mehrere Sende- und Empfangsspeicher.

**Initialisierungswert** Wert eines Signals vor dem ersten Empfang im Empfänger.

**Knoten** Ein Netzknoten umfaßt jeglichen Teilnehmer oder auch Abzweig des Netzwerks.


**Latenzzeit** Übertragungsparameter einer Botschaft, der die Übertragungszeit auf dem Netz beschreibt. Die Latenzzeit ist die Zeit, um der Sendevorgang einer Botschaft durch höherpriorie Botschaften verzögert werden kann. Technisch gesprochen ist die Latenzzeit die Zeitspanne, die zwischen TransmitRequestBit setzen und AckMeldung erhalten liegt. An der Botschaft kann also die maximale Latenzzeit für diese Botschaft spezifiziert werden.

**Multiplex-Signal-Gruppe** In einer Multiplex-Signal-Gruppe innerhalb einer Botschaft werden mehrere Signal-Gruppen zusammengefaßt, die über einen jeweiligen (Gruppen-)Wert identifiziert werden. Ein Empfänger der Botschaft kann die für ihn bestimmte Signal-Gruppe über diesen Wert herausfiltern.

**Net-EMV-Design** Beschreibung des Designs bzw. der Auslegung des Netzwerkes hinsichtlich seiner elektromagnetischen Verträglichkeit.

**Netz-Signal-Gruppe ...**

**Netz-Signal-Klasse** Die Netz-Signal-Klasse teilt die Netz-Signale in Applikationssignale und Netzwerkmanagementsignale ein.

	Strukturelle Grundlagen SNT	Seite: 54/75
	Kapitel: Glossar	Datum: 2002-02-07 Status: RD

**OSEK** To be defined

**Phasenbeziehung zu anderen Botschaften** Übertragungsparameter einer Botschaft, der beschreibt, ob eine Phasenbeziehung zu anderen Botschaften vorliegt.

**Physikalischer Bereich** Zulässiger Bereich für die physikalischen (externen) Werte eines Signals, die sich aufgrund der Umrechnung von der Netz-internen Darstellung ergeben.

**Quarzfrequenz** Der Frequenzwert des am CAN-Controller angeschlossenen Quarzes bzw. Quarzoszillators.

**RTR** Abkürzung für Remote Transmission Request; ein Flag zur Kennzeichnung einer Datenanfrage. Ein Sender wird aufgefordert ein entsprechendes Datentelegramm als Antwort zu senden.

**Sendebedingungen** Übertragungsparameter einer Botschaft, der Sendebedingungen beschreibt.

**Sendebereitschaft nach Power up** Übertragungsparameter einer Botschaft, der den Zeitpunkt nach *Power up* beschreibt, ab wann die Botschaft gesendet werden kann.

**SJW** Abkürzung für Synchronization Jump Width, die in BTL-Zyklen angegeben wird. Definiert den maximalen Betrag, um den der CAN-Controller die beiden Hauptsegmente der Bitzeit verlängern oder verkürzen darf.

**Sleep-/Wake up-Mechanismus** Ein Mechanismus, der den Netz in einen " *Ruhezustand*" mit geringer Stromaufnahme versetzt (Sleep) und ihn bei Bedarf wieder aktiviert (Wake up); z.B. für eine Zentralverriegelung.


**SLIO** Abkürzung für Serial Linked I/O.

**Speicherlayout** Für eine Botschaft ist das Speicherlayout eine (grafische) Ergänzung zu der Beschreibung von Signallisten und Multiplex-Signal-Gruppen, die die Signalverteilung einer Botschaft im Speicher beschreibt.

**Umrechnung** Vorschrift, wie die Netz-interne Darstellung (interner Werte) eines Signals in die zugehörigen physikalischen Werte (externe Werte) umzurechnen sind.

**Vorteiler** oder auch Baudrate Prescaler; zur Erzeugung des BTL-Zyklen-Taktes aus der Quarzfrequenz.

**Zykluszeit** (Auch Zyklustoleranzzeit), Übertragungsparameter einer Botschaft, der die Toleranzzeit einer Botschaft bei zyklischer Steuerung (zyklisches Versenden der Botschaft) beschreibt.

	Strukturelle Grundlagen SNT	Seite: 55/75
	Kapitel: Literaturverzeichnis	Datum: 2002-02-07 Status: RD

## Anh. B Literaturverzeichnis

*[Externes Dokument: CAN-ER-Modell / Datum: 1988 / Herausgeber: Robert Bosch GmbH / URL: / relevante Stelle: ]*

*[Externes Dokument: Analyse zur CAN-Datenbank, Unterlage der Fa.Bosch, / Dokumentnummer: / Status: / Datum: 17.04.1989 / URL: / relevante Stelle: ]*

*[Externes Dokument: CAN-Controller Area Network, Grundlagen und Praxis / Datum: 1984 / Herausgeber: Hüthing Verlag / URL: / relevante Stelle: ]*

*[Externes Dokument: CAN Controller Area Network, Grundlagen, Protokolle, Bausteine, Anwendungen / Datum: 1994 / Herausgeber: Carl Hanser Verlag München Wien, München / URL: / relevante Stelle: ]*

## Anh. C Legende der Strukturgrafiken

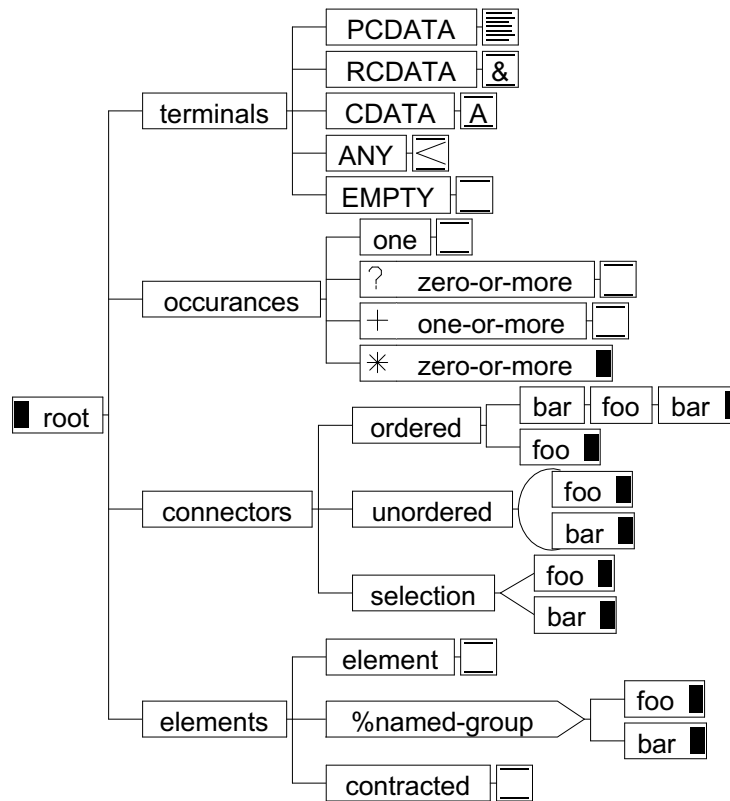


Abbildung 14: Bedeutung der Symbole in den Strukturgrafiken

legend.gif



## Anh. D Betrachtete Alternativen zur Ankopplung von SG und Netz

### Anh. D.1 Kopplung über Einzelsignale

Im Fall a) werden die Einzelsignale der Netzverbindung (z.B. *Low*, *High*, *Shield*) spezifiziert und bei der Verbindungsspezifikation auf die entsprechenden Ports gelegt.

Bei b) wird nur ein Netz-Signal definiert und auf alle Ports gelegt.

### Anh. D.2 Verwendung von Namenskonventionen

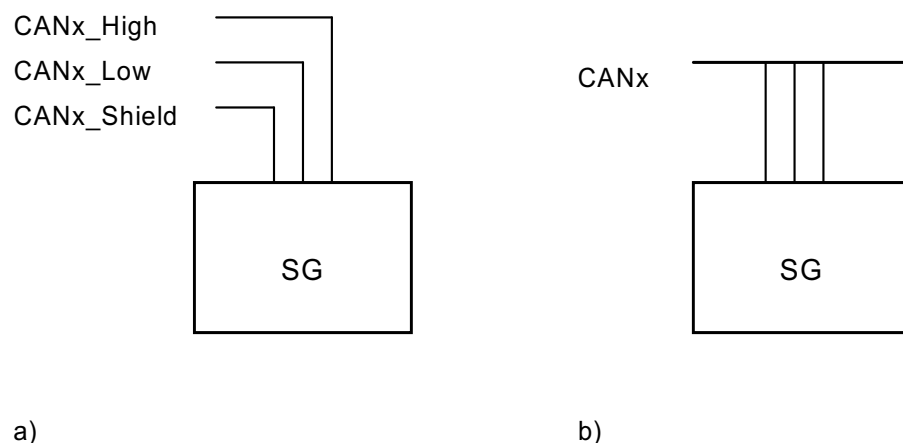
Die Namen der Einzelleitungen des Netzes werden wie normale Signale behandelt. Die Zuordnung der Steuergeräte-Signale zu einem Netz erfolgt über eine Namenskonvention, indem die Bezeichnung des Netzes als Präfix im Signalnamen wieder auftritt. Für eine Netz CANx könnten die Signale die Namen CANx\_High, CANx\_Low, und CANx\_Shield tragen.

Diese Variante ist nicht ausreichend, da im Fall c) in [Abbildung 2 Kopplung von Steuergerät und Netz S. 14](#) nicht eindeutig ist an welche Herausführung das Netz aufgelegt wird.

Diese Namenskonventionen können auch dadurch realisiert werden, daß Signale in **< signal >** ein weiteres Element **< net-name >** bekommen, welches die Einzelsignale dem Netz zuordnet.

### Anh. D.3 Einführung eines Netz-Signalelementes

Zusätzlich zu der bestehenden Signalbeschreibung wird eine neue Art der Signaldefinition eingeführt. Dieses kann innerhalb der Signalspezifikation **< signal-spec >** oder innerhalb einer Signalgruppe **< signal-group >** erfolgen.



a) b)  
 Abbildung 15: Zuordnung der Netzleitungen zum Steuergerät (SG)

```

a)
<signal-group>
  <signals>
  <net-lines>
    <long-name>
    <short-name>
    <variant-def-refs>
  
```

```

b)
<signal-spec>
  <signal-groups>
  <net-lines>
  
```

- a) **<signals>** und **<net-lines>** sind insoweit äquivalent, als daß bei **<net-lines>** noch Mittel zur Bezeichnung des Netzes vorhanden sind. Daß **<signals>** in **<net-lines>** enthalten ist, liegt an der Konvention, daß mehrfach auftretende Elemente ein Hüllenelement haben sollten. Rein logisch betrachtet, kommt hier eine Menge von **<signal >** vor wie in **<signals >** auch. Bei dieser Lösung müssen für ein Netz z.B. drei **<signal-group>** angelegt werden.

```

<signal-group>
  <signals>
  <net-lines>
    <long-name>           Bezeichnung des Netzes
    <short-name>         Bezeichnung des Netzes
    <signals>
  <signal-spec-variants>
  
```

- b) Dieser Ansatz ist äquivalent zu a). Auch hier sind drei **<signal-group >** anzulegen. Die Hierarchie ist jedoch etwas flacher.

```


<signal-group>
  <signals>
  <net-lines>
    <net-line>
      <long-name>       Signalbezeichnung
      <short-name>     Signalbezeichnung
      <net-name>       müßte konsequenterweise <net-ref>
                      sein,
                      ist im Grunde ein Zeiger aus der
                      MSRDOC-Instanz in MSRNET Instanz.
    <variant-def-refs>
  
```

- c) Diese Variante beschreibt das Netzwerk geschlossen. Auch hier werden für ein Netzwerk drei **< signal-group>** Elemente angelegt, die jedoch unter einem zusammengefaßt sind. Diese Variante hat den Nachteil, daß Signale an verschiedenen Stellen definiert werden. Wenn also alle Signale einer Komponente aufgesucht werden sollen, muß an mehreren Stellen gesucht werden. Dem steht der Vorteil gegenüber, daß ein Netzwerk in sich komplett beschrieben wird.

```

<signal-spec>
  ...
  <busses>
    <bus>
      <long-name>       Busbezeichnung
      <short-name>
      <signal-groups>
  
```

Diese Variante adressiert ebenfalls nicht das Problem [Abbildung 2 Kopplung von Steuergerät und Netz S. 14 Fall b\)](#). Um das zu behandeln, wären weitere Elemente notwendig, welche den Signalnamen modifizieren.

	Strukturelle Grundlagen SNT	Seite: 59/75
	Kapitel: Referenzieren von Netzelementen bei der Signaldefinition	Datum: 2002-02-07 Status: RD

## Anh. D.4 Referenzieren von Netzelementen bei der Signaldefinition

Netzsignale werden wie Signale behandelt. Die Kennzeichnung als Netzsignal erfolgt über eine Referenz auf ein neues Netzelement.


```
<architecture>
  <buses>
    <bus>
      1
      2  <-
        |
<signal> -> <bus-ref>
```

Innerhalb der Architecture werden Netzwerke eingeführt. Jedes Signal kann wahlweise genau einem Netzwerk zugeordnet werden. Dies ist eine Fortentwicklung von [Topic Anh. D.3 Einführung eines Netz-Signalelementes S. 57](#), indem das dortige **<net-name>** normiert wird.

```
<architecture>
  <buses>
    <bus>
      <long-name>
      <short-name>
      <desc>
      <add-info>

<signal>
  <bus-ref>
```

Löst noch nicht die Problematik [Abbildung 2 Kopplung von Steuergerät und Netz S. 14 Fall b\)](#).

	Strukturelle Grundlagen SNT Kapitel: net-signal-class	Seite: 60/75 Datum: 2002-02-07 Status: RD
---	---	---

## **Anh. E Festlegung von Wertebereichen**

### **Anh. E.1 node-type**


TBD

### **Anh. E.2 topology-type**

TBD

### **Anh. E.3 net-signal-class**

TBD

	Strukturelle Grundlagen SNT  Kapitel: Zusätzliche Anmerkungen zum Dokument- stand 1.1.1 am 10.7.98	Seite: 61/75 Datum: 2002-02-07 Status: RD
---	--	---

## **Anh. F      Zusätzliche Anmerkungen zum Dokumentstan- d 1.1.1 am 10.7.98**

Status dieses Dokument: CD

## Dokumentverwaltung

### Änderungs- übersicht

Gesamt	Dokumentteil	Nr.	Änderung	Grund	Bezug
30.05.96		11	Umsetzung nach SGML, Redaktionelle Überarbeitung	Qualitätsverbesserung	Dokument
			Konvertierung nach SGML	AG-Beschluß	Dokument
			Einarbeitung der Sitzungsergebnisse vom 13.3.96	AG-Beschluß	Inhalt
			Bild- und Tabellenüberschriften entfernt	Diese werden vom Formattierer erzeugt.	Dokument

Gesamt	Dokumentteil	Nr.	Änderung	Grund	Bezug
05.06.96		10	Grundsätzliche Überarbeitung der Struktur	Abdeckung verschiedener Netztopologien und Netztypen - Vereinheitlichung mit MSR-SW	Inhalt
			Die Ankopplung von NET und SG wurde grundsätzlich überarbeitet. Kapitelstruktur eingefügt	Übersichtlichkeit, und Abdeckung 2b	Inhalt
			Bus wurde zu Net umbenannt.	Es werden nicht nur Bustopologien beschrieben.	Dokument
			Bus-Instance wurde zu Net-Port	Weil es keine Businstanzen gibt	Inhalt
			Hinweise zum Netzmanagement wurden überarbeitet (s. <a href="#">Topic 3.3.1 Allgemeines Netzwerkmanagement S. 26</a> )	Aktualisierung	Inhalt
			Blockmode hat keine eigene Struktur mehr (s. <a href="#">Topic 3.3.5 Blockübertragungsmodi S. 27</a> )	Die Botschaften der Blockübertragungsmodi werden können auch durch normale Botschaften beschrieben werden.	Inhalt
			SW-Compu-Method wurde übernommen	Vereinheitlichung	Inhalt
			Physikalische Eigenschaften bei den Netzsignalen optional eingeführt.	Unterstützung von Blockmode und sonstigen Signalen	Inhalt
			Glossar wurde aktualisiert	Update	Inhalt
			Sende-/Empfangsbereitschaften wurden optional	Diese Daten treten primär nur bei Netzwerkmanagementsignalen auf	Inhalt
			Der Multiplexor wurde explizit eingeführt.	Erhöhung der Transparenz für den User	Inhalt
Offsets werden grundsätzlich auf Botschaftanfang bezogen (s. <a href="#">Topic 3.3.7 Botschaften S. 29</a> ), Multiplexor- Signal-Gruppe verschachtet.	Erhöhte Flexibilität	Inhalt			
Kapitel Varianten wurde eingeführt (s. <a href="#">Topic 3.4 Varianten S. 34</a> )	Vorbereitung der Variantenbehandlung.	Dokument			
12.06.96		9	BUS auf NET geändert	Bereinigung des Berichtes	Inhalt

Gesamt	Dokumentteil	Nr.	Änderung	Grund	Bezug
22.08.96		8	Konvertiert auf dtd16.6	Qualitätsverbesserung	Dokument
			<a href="#">Topic 3.2.2.1 Netzleitungen S. 21</a> : Abschlußwiderstand durch Abschlußbeschaltung ersetzt	Netze können komplexere Abschlußbeschaltungen haben	Inhalt
			<a href="#">Topic 3.2.2.1 Netzleitungen S. 21</a> : Struktur von Knoten durch Knotenport ergänzt	Abdeckung von Fall 2b)	Inhalt
			<a href="#">Topic 3.2.2.1 Netzleitungen S. 21</a> : Knotennummer durch Knotenname ersetzt (short-name)	Knotenkurzbezeichnung ist nicht nur numerisch	Inhalt
			<a href="#">Topic 3.2.1 Verbindungskomponenten S. 19</a> : Definition der Verbindungskomponenten erfolgt nun innerhalb einer MSRNET-Instanz	Damit werden Netz-Beschreibungen "self-contained"	Inhalt
			<a href="#">Topic 3.3.6 Netz-Signale S. 27</a> : Referenz auf das physikalische Signal entfernt	Vermeidung dokumentübergreifender Verweise. Diese Referenz wäre semantisch unklar.	Inhalt
			Umrechnungsformeln werden innerhalb MSR-NET definiert	MSRNET-Instanzen werden self-contained - Umrechnungsformeln können als Entity eingezogen werden.	Inhalt



Gesamt	Dokumentteil	Nr.	Änderung	Grund	Bezug
09.10.96		7	Redaktionelle und Strukturelle Verbesserungen (Fehlerbehandlung, Abschlußwiderstand)	Rückmeldung aus hausinternen Reviews	Dokument
			Redaktionelle Überarbeitung, Tippfehlerbereinigung	Verbesserung der Lesbarkeit	Dokument
			<a href="#">Firma 2.1 Anforderungen S. 13</a> in <a href="#">Topic 2 Zuordnung von Netz und Steuergerät S. 13</a> eingefügt	Verbesserung der Struktur	Dokument
			Einführung von networkspec in <a href="#">Topic 2.2 Beschreibung von Netzwerk-Anschlüssen in MSRDOC S. 14</a> detailliert begründet	Verbesserung der Klarheit	Dokument
			DTDs umbenannt in <a href="#">Abbildung 1 MSR-DOC.DTD und MSR-NET.DTD S. 11</a>	Fehlerkorrektur	Dokument
			<a href="#">Topic 3.3.3 Fehlerbehandlung S. 27</a> eingeführt	Sicherstellen, daß Fehlerbehandlung explizit bearbeitet wird und nicht in der allgemeinen Beschreibung untergeht.	Inhalt
			Abschlußwiderstand auch in Struktur durch Interface-circuit ersetzt in <a href="#">Topic 3.2.2.1 Netzleitungen S. 21</a>	Fehlerbehebung	Inhalt

Gesamt	Dokumentteil	Nr.	Änderung	Grund	Bezug
21.04.97		6	Redaktionelle Überarbeitung	Verbesserung der Lesbarkeit	Dokument
			Erneuerung der Grafiken	Grafiken waren undeutlich	Inhalt
			Entfernung des Kapitels "Alternative Netzbeschreibungskonzepte"	Kapitel war unverständlich	Inhalt
			Verbesserung der Darstellung, Verfeinerung der Beschreibungen von Netports, net-interface, Block-Modi, Netz-Signale, Botschaften	Vervollständigung um weitere Aspekte von Netzen, Klärung von Details	Inhalt
			Elementnamen eingeführt	Synchronisierung mit DTD-Entwicklung	Inhalt
			Anmerkungen zur Gateway-Funktionalität	Klärung	Inhalt
			Referenzierung der Netports um Referenz auf Part erweitert	Netze werden mit Parts verbunden, nicht mit part-types	Inhalt
			Beschreibung net-interface detailliert	Beschreibung aller Aspekte des Netzes	Inhalt
			Beschreibung der Block-Modi eingeführt	Beschreibung aller Aspekte des Netzes	Inhalt
			Netz-Signale und Netz-Signal-Gruppen <a href="#">Topic 3.3.6 Netz-Signale S. 27</a> detailliert	Klärung	Inhalt
			Aufbau der Botschaften systematisiert	Vereinfachung und Klärung	Inhalt
05.11.97		5	Anforderungen aus Informationsgespräch bei BMW eingetragen		Inhalt
18.11.1997		4	Ergebnisse der AG-NET 18.11.97 eingetragen	Vorgabe für Implementierung 1.1.0	Inhalt
18.02.1998		3	Redaktionelle Überarbeitung des Berichtes	Dokumentation für 1.1.1	Inhalt
			Einbau der Änderungen in Changes für 1.1.1	Detaildiskussion zeigte die Notwendigkeit	Inhalt
			Zusammenfassung von Botschaften zu Dokumentierungszwecken.	Klärung	Inhalt
1998-07-10		2	Umgesetzt auf neue MSR-MEDOC Dokumentstrategie (MSR-TR-DOV)	neue Strategie	Inhalt
2002-02-07		1	Erstellen von Indexen, Technischen Begriffen und Querverweisen. Konvertieren nach MSR-REP V210 XML.	Aufbereitung für Abschlussdokumentation	Inhalt

**Versions-  
übersicht**

Dokument- teil	Datum	Herausgeber			
		Firma	Version	Status	Anmerkun- gen
ab Seite 8	2002-02-07	Dipl.-Ing. R. Reimer			
	Änderungen 1	MEDOC	1.1.2	RD	
	1998-07-10	Dipl.-Ing. B. Weichel			
	Änderungen 2	MEDOC	1.1.1	CD	
	18.02.1998	Dipl.-Ing. B. Weichel			
	Änderungen 3	MEDOC	1.1.1	abgeschlos- sen	
	18.11.1997	Dipl.-Ing. B. Weichel			
	Änderungen 4	MEDOC	11.0f	abgeschlos- sen	
	05.11.97	Dipl.-Ing. B. Weichel			
	Änderungen 5	MEDOC	11.0b	outdated	
	21.04.97	Dipl.-Ing. B. Weichel			
	Änderungen 6	MEDOC	11.0a	outdated	
	09.10.96	Dipl.-Ing. B. Weichel			
	Änderungen 7	MEDOC	10.3	fertig	
	22.08.96	Dipl.-Ing. B. Weichel			
	Änderungen 8	MEDOC	10.2	abgeschlos- sen	
	12.06.96	Dipl.-Ing. B. Weichel			
	Änderungen 9	MEDOC	10.1	überholt	
	05.06.96	Dipl.-Ing. B. Weichel			
	Änderungen 10	MEDOC	10.0	überholt	
	30.05.96	Dipl.-Ing. B. Weichel			
	Änderungen 11	MEDOC	9.1	Nach Sitzung vom 13.3.96	

## Literaturverzeichnis

### Normen

<b>Bezeichnung:</b>	[ISO 11519-1]: To be defined	
<b>Status:</b>	DIS	
<b>URL:</b>		
<b>relevante Stelle:</b>	gesamtes Dokument	<a href="#">52</a>

<b>Bezeichnung:</b>	[ISO 11898]: To be defined	
<b>Status:</b>	DIS	
<b>URL:</b>		
<b>relevante Stelle:</b>	ganzes Dokument	<a href="#">52</a>

### Externe Dokumente

<b>Bezeichnung:</b>	Analyse zur CAN-Datenbank, Unterlage der Fa.Bosch,	
<b>Dokumentnummer:</b>		
<b>Status:</b>		
<b>Datum:</b>	17.04.1989	
<b>URL:</b>		
<b>relevante Stelle:</b>	alles	<a href="#">55</a>

<b>Bezeichnung:</b>	CAN Controller Area Network, Grundlagen, Protokolle, Bausteine, Anwendungen	
<b>Datum:</b>	1994	
<b>Herausgeber:</b>	Carl Hanser Verlag München Wien, München	
<b>URL:</b>		
<b>relevante Stelle:</b>	ganzes Buch	<a href="#">55</a>


<b>Bezeichnung:</b>	CAN-Contoller Area Network, Grundlagen und Praxis	
<b>Datum:</b>	1984	
<b>Herausgeber:</b>	Hüthing Verlag	
<b>URL:</b>		
<b>relevante Stelle:</b>		<a href="#">55</a>

<b>Bezeichnung:</b>	CAN-ER-Modell	
<b>Datum:</b>	1988	
<b>Herausgeber:</b>	Robert Bosch GmbH	
<b>URL:</b>		
<b>relevante Stelle:</b>	handschriftliche Grafiken und SQL-Strukturdefinitionen	<a href="#">55</a>

<b>Bezeichnung:</b>	ISO Norm CAN High-Speed	
<b>URL:</b>		
<b>relevante Stelle:</b>	kpl.	<a href="#">24</a>

<b>Bezeichnung:</b>	ISO Norm CAN Low-Speed	
<b>URL:</b>		
<b>relevante Stelle:</b>	kpl.	<a href="#">24</a>

<b>Bezeichnung:</b>	Strukturelle Grundlagen, Bereich Software	
<b>Status:</b>	1.1.0	
<b>Datum:</b>	16.12.97	
<b>Herausgeber:</b>	MSR AG-MEDOC	
<b>URL:</b>		
<b>relevante Stelle:</b>	Kapitel Inhaltsmodell Software	<a href="#">29</a>

	Strukturelle Grundlagen SNT Kapitel: Index	Seite: 70/75 Datum: 2002-02-07 Status: RD
---	--	---

## Index

### **B**

Botschaften [29](#)

### **K**

Knotenart [22](#)

### **M**

MEDOC [2](#)

### **N**

Netz-Formate [10](#)

Netz-Interface [24](#)

Netztopologie [12](#)

Normen [9](#)

### **S**

Signale [26](#), [59](#)

### **V**

Varianten [34](#)

## Technische Begriffe

### Organisationen

### Symbole

ASAP [27](#)

ISO TC22 SC3 WG1 [27](#)

OSEK [27](#)

### M

MEDOC-Netzwerke [10](#)

### Parameter

### L

Latenzzeit [19](#)

### Z

Zykluszeit [19](#)

### Produkte

### Symbole

ABS-Steuergerät [11](#)

CAN-controller [25](#)

Motor-Steuergerät [11](#)

MSRNET.DTD [25](#), [41](#)

MSRSW.DTD [41](#)

Steuergerät [11](#), [11](#), [14](#), [14](#)

### F

Fahrzeugsystem [12](#)

### M

MSRDOC [24](#)

MSRDOC.DTD [11](#), [12](#), [14](#), [22](#), [22](#)

MSRNET.DTD [9](#), [12](#), [15](#), [37](#), [46](#)

MSRSW.DTD [41](#), [50](#), [51](#)

MSRSYS.DTD [47](#)

MSWDOC.DTD [41](#)

### N

Netzwerke [2](#), [9](#)

### S

Steuergerät [11](#), [13](#), [13](#), [14](#), [25](#), [31](#), [45](#)

Steuergerät [14](#)

### Programmcode

### E

extended [38](#), [45](#)

### S

standard [38](#), [45](#)

### SGML Elemente

### Symbole

abs [25](#)

add-info [26](#)

admin-data [18](#)

company-doc-info [49](#)

connection-comp-spec-1 [19](#)

cycle-mode [44](#), [44](#)

driver-concept [25](#)

identifizier-type [38](#)

interface-circuit [34](#)

long-name [22](#), [30](#)

msg-identifizier-offset [30](#), [34](#), [48](#)

multiplex-signal-set [31](#)

multiplexor [31](#)

na [50](#)

net-architecture [19](#)

net-emc-design [24](#)

net-interface-prms [24](#)

net-line [21](#)

net-lines [58](#)

net-message [34](#), [34](#), [48](#)

net-message-group [48](#)

net-message-set [37](#), [47](#)

net-message-signals [32](#)

net-name [57](#)

net-node [34](#)

net-node-port [22](#), [22](#), [49](#)

net-node-port-ref [22](#)

net-port-variant [35](#)

net-signal [34](#)

net-signal-class [27](#)

net-signal-spec-variant [37](#), [47](#)

net-topology-spec [19](#)

part-ref [22](#)

segment [34](#)

short-name [21](#)

signal [15](#), [57](#)

signal-group [57](#), [58](#)

signal-spec [57](#)

sw-base-type [37](#)

sw-compu-method-value-pair [45](#)

sw-units [27](#)

**A**

abs [25, 25](#)  
add-info [24](#)  
add-spec [19](#)  
address-mode [31](#)  
admin-data [35, 49, 49](#)  
adress-mode [38, 45](#)  
architecture [14](#)

**B**

baudrate [25, 25](#)  
bitsize [28, 31](#)  
btl-cycles [25, 25](#)  
byte-order [31](#)

**C**

calc-net-message-identifiers [30, 34, 48](#)  
cmt-desc [45](#)  
cmt-text [45](#)  
coded [28](#)  
connection-comp-ref [22, 22](#)  
connection-comp-spec-1 [19](#)  
cycle-mode [36](#)  
cycle-time [33](#)

**D**

desc [35, 45](#)  
dlc [31](#)  
driver-concept [24](#)

**E**

error-value [28](#)  
error-values [28](#)  
event-mode [36, 44, 44](#)

**G**

general-net-spec [18, 19](#)

**I**

identifier [30](#)  
identifier [31](#)  
Identifier-base-adress [30, 34, 48](#)  
identifier-type [45](#)  
identifier-type [45](#)  
init-value [28, 37, 47, 47](#)  
interface-circuit [22, 22](#)

**L**

latency-time [33](#)  
locs [51](#)  
long-name [18, 21](#)

**M**

memory-layout [37](#)  
multiplex-entry [31](#)  
multiplex-signal-list [31](#)  
multiplexor-value [31](#)

**N**


na [50](#)  
net-architecture [18](#)  
net-block-modes [27](#)  
net-diag-spec [27](#)  
net-emc-design [25](#)  
net-error-handling [27](#)  
net-init-spec [27](#)  
net-interface-prms [25](#)  
net-interface-spec [24](#)  
net-interface-spec [19](#)  
net-lines [21, 58](#)  
net-lines [58](#)  
net-message [26, 37](#)  
net-message [30, 30](#)  
net-message-desc [30](#)  
net-message-identifiers [30, 34, 48](#)

net-message-layout [32](#)  
net-message-layout [32, 37](#)  
net-message-set [29, 37](#)  
net-message-signal [31, 31, 34](#)  
net-message-signals [31, 31, 32, 32](#)  
net-message-signals [29, 31](#)  
net-message-spec [26, 29, 29](#)  
net-mgmt-spec [26](#)  
net-name [59](#)  
net-node-port [22](#)  
net-node-variant [49](#)  
net-nodes [20](#)  
net-oper-spec [18](#)  
net-oper-spec [26](#)  
net-port-ref [22](#)  
net-sig-var-assocs [51](#)  
net-signal [26, 31, 43](#)  
net-signal-class [28, 28](#)  
net-signal-group [34](#)  
net-signal-group [28](#)  
net-signal-spec [26](#)  
net-signal-spec [27](#)  
net-signal-spec-variant [37, 43, 44](#)  
net-signal-spec-variant [28, 44](#)  
net-toplogy-spec [20](#)  
netline-desc [21](#)  
network-spec [14, 15](#)  
node-type [22](#)  
node-variant [38, 46](#)

**O**

offset [31, 31](#)



	Strukturelle Grundlagen SNT	Seite: 73/75
	Kapitel: Status	Datum: 2002-02-07 Status: RD

## P

part-type-spec [24](#)  
phase-relations [33](#)  
phys [28](#)  
port-groups [43](#)  
powerdown-receive-time [32](#)  
powerup-receive-time [32](#)  
powerup-transmit-time [32](#)  
project-data [18](#)  
protocol-conformance [46](#)  
protocol-conformance [38](#)

## R

receivers [29](#), [32](#)  
role [49](#)

## S

sample-point [25](#), [25](#)  
sample-rate [25](#)  
segment [22](#)  
segment-length [22](#)  
segmentation [20](#)  
segmentation-desc [22](#)  
segments [22](#)  
sender [31](#), [32](#)  
sender [30](#), [34](#), [48](#)  
setmentation [50](#)  
short-name [18](#), [30](#), [37](#)  
short-name [21](#), [37](#)  
signal [58](#)  
signal-class [15](#)  
signal-group [58](#)  
signal-group [58](#)  
signals [58](#), [58](#)  
signals [58](#)  
sjw [25](#)

sjw [25](#)

sw-base-type [44](#)  
sw-compu-method-ref [29](#)  
sw-compu-method-text [35](#), [45](#)  
sw-compu-method-value-pair [35](#)  
sw-compu-methods [27](#)  
sw-compu-methods [29](#)  
sw-limits [28](#)  
sw-variable [43](#), [44](#)  
sync-edge [25](#)

## T

Team-Member [49](#)  
team-member-ref [49](#)  
text [25](#)  
tol [25](#), [25](#), [25](#)  
topology-type [20](#)  
transmission-spec [32](#), [34](#), [38](#), [50](#)

## V

variant-def-ref [31](#)  
variant-def-refs [22](#)

## X

xref [34](#)

## Sonstige

### Symbole

Applikationssignal [28](#)  
CAN\_LOW [21](#)  
Low [57](#)  
Netz-signalgruppe [27](#)  
Netzwerkmanagementsignal [28](#)  
Power down Meldung [32](#), [53](#)  
Standard-Format [31](#), [45](#)

## C

CAN\_HIGH [21](#)  
CAN\_SHIELD [21](#)

## E

Extended-Format [31](#), [45](#)  
Extended-Format [31](#), [45](#)

## H

High [57](#)

## S

Shield [57](#)

## Status

### Symbole

Power up [32](#), [54](#)  
Ruhezustand [54](#)  
Teilnehmer [22](#)

## A

aktiv [31](#), [45](#)

## H

Hilfsknoten [22](#)

## I

intel [31](#)

## M

motorola [31](#)  
motorola-rueckwaerts [31](#)  
motorola-vorwaerts [31](#)



## **P**

passiv [31](#), [45](#)

Power up [32](#)

## **S**

STD [31](#)

## **X**

XTD [31](#)

## **Variablen**

### **R**

Registerbelegung [25](#)

## **Werkzeuge**

### **Symbole**

MSRDOC.DTD [11](#), [11](#)

MSRNET.DTD [11](#), [15](#)

## **C**

CAN [10](#)

### **M**

MSRDOC.DTD [15](#), [21](#), [22](#), [22](#)

MSRDOC.DTD [19](#)

MSRNET.DTD [12](#)

### **S**

SGML-Formatierer [32](#)

## Konfigurationsparameter

**Unternehmen (—company)**

MEDOC

**Sprache (—lang)**

Deutsch

**Ausgeben des Inhalts von Querverweisen (—xrefcontent)**

Inhalt von Xref wird ausgegeben

**Ausgabe von 'Siehe' für Querverweise**

'Siehe' wird nicht ausgegeben

**Dateinamenausgabe für Grafiken (—figname)**

Dateinamen für Grafiken werden ausgegeben

**Verarbeitung von [width] und [height] Attribut für Grafiken (—figdimension)**

Breite und Höhe von Grafiken werden verarbeitet

**Titelseiten Graphik (—graphic)**

Keine Titel Grafik spezifiziert

**Logo Graphik (—head-logo)**

msrreportlogo.eps

**Fixtexte Datei (—fixtext)**

C:\Programme\Medoc\Metapage\mmapps\msrrep\lib\msrrep\_ft.xml

**Ausgabe der lokalen administrativen Daten (—admindata)**

Lokale administrative Daten werden ausgegeben

**Datei**

C:\Docs\MEDOC\MSRNET\docs\de\sp\msrnet-sp-de\_V210.xml

**MetaMorphosis-Version**

3.2

**Formatierer-Version**

2.0 (MetaPage)

**Datum**

11/02/2002 12:17:41