



Datenaustausch zwischen IQ-FMEA und anderen Systemen im XML/SGML-Format

MSR MEDOC, Herbert Klein, XI-Works



Inhaltsverzeichnis

	Inhaltsverzeichnis	2
	Präliminarien	3
1	Einleitung	4
2	Motivation	5
3	Das MSR-Projekt	7
4	Was ist XML/SGML	8
4.1	Entwicklungsgeschichte	8
4.2	Anwendungsbereiche	8
4.3	L-Grundbegriffe	9
4.4	Bestandteile eines SGML-Dokumentes	11
4.4.1	Die SGML-Deklaration	12
4.4.2	Die DTD eine baumartige Dokumentstruktur-Definition	12
4.4.3	Die SGML-Instanz	12
5	MSR-Standards	16
5.1	Das SGML Applikationsprofil von MSR MEDOC	16
5.2	Die einzelnen MSR-Standards	16
6	Die MSR FMEA DTD	18
6.1	Ziele der MSRFMEA.DTD	18
6.2	Prozeßmodelle für die DTD	18
6.2.1	Hinzuladen von Maßnahmen bzw. Funktionskatalogen	18
6.2.2	Manipulation der FMEA mit anderen Werkzeugen	19
6.2.3	FMEA-Erstellung auf Basis von Engineering-Tool Daten	19
6.2.4	Wiederherstellen einer FMEA aus der SGML-Instanz	19
6.2.5	Modularisierung	19
6.3	Grobstruktur der DTD	19
7	Literaturverzeichnis	22
8	Referent	23
	Dokumentverwaltung	24



Präliminarien

Projektbeteiligte
Firmen

MSR MEDOC [MEDOC]

Name Rollen	Abteilung	Adresse	Kontakt
Herbert Klein Autor	XI-Works	Olgastr. 86 D-70180 Stuttgart	Telefon +49 711 248398-10 FAX +49 711 248398-29 Herbert.Klein@xi-works.de www.xi-works.de
Roman Reimer, XI-Works Umsetzung in XML	XI-Works		

Versionsinformation

Dokumentteil	Herausgeber			
	Firma	Version	Status	Anmerkungen
1 RD 1999-10-12 Details siehe Nr. 1, Seite	Herbert Klein			
	MEDOC			



1 Einleitung

Im Rahmen einer Zusammenarbeit zwischen dem MSR-Teilprojekt **MEDOC** (MSR-Entwicklungsdokumentation) und der Firma TKB entstand eine Austauschdatenstruktur auf Basis der SGML/XML-Technologie.

2 Motivation

Die zunehmende Interaktion im Entwicklungsprozeß zwischen Fahrzeugherstellern und Elektronikzulieferern bringt auch eine Zunahme des Daten- und Dokumentaustauschs mit sich. Dabei werden in den unterschiedlichen Unternehmen auch unterschiedliche IT-Werkzeuge eingesetzt, weshalb ein standardisiertes Datenaustauschformat benötigt wird.



Abbildung 1: Prozeßinteraktion Hersteller/Zulieferer

Die gesuchten Formate haben dabei die Aufgabe, sowohl den Austausch von (formalen) Datenbank- und CAD-Daten, als auch den Austausch von (informellen) Textdokumenten zu organisieren. Außerdem ist die Aufgabe zu lösen, eine Gesamtdokumentation, bestehend aus den formalen und informellen Anteilen, erstellen zu können. Folglich benötigt man auch eine Technologie zur Dokumentorganisation.

In der Vergangenheit wurden deshalb in einer Workshop-Reihe zunächst verschiedene Dokumentaustausch-Formate untersucht und gegenübergestellt. Ein Kandidat war der Dokumentaustausch-Standard SGML (Standard Generalized Markup Language ISO 8879). Eine genauere Betrachtung dieser Technologie sowie existierender SGML-Anwendungen in der Automobil-, Elektronik- und Luftfahrtindustrie ergab, daß mit Hilfe dieses Formats neben der Lösung von Dokumentations- und Publikationsaufgaben auch der Austausch von Daten organisiert werden kann. Diese Chance, alle anstehenden Aufgaben auf Basis eines einzigen Formats lösen zu können, sowie die Verfügbarkeit von SGML-Werkzeugen am Markt waren ausschlaggebend für die Wahl von SGML als Basistechnologie für den Daten- und Dokumentaustausch in MSR.

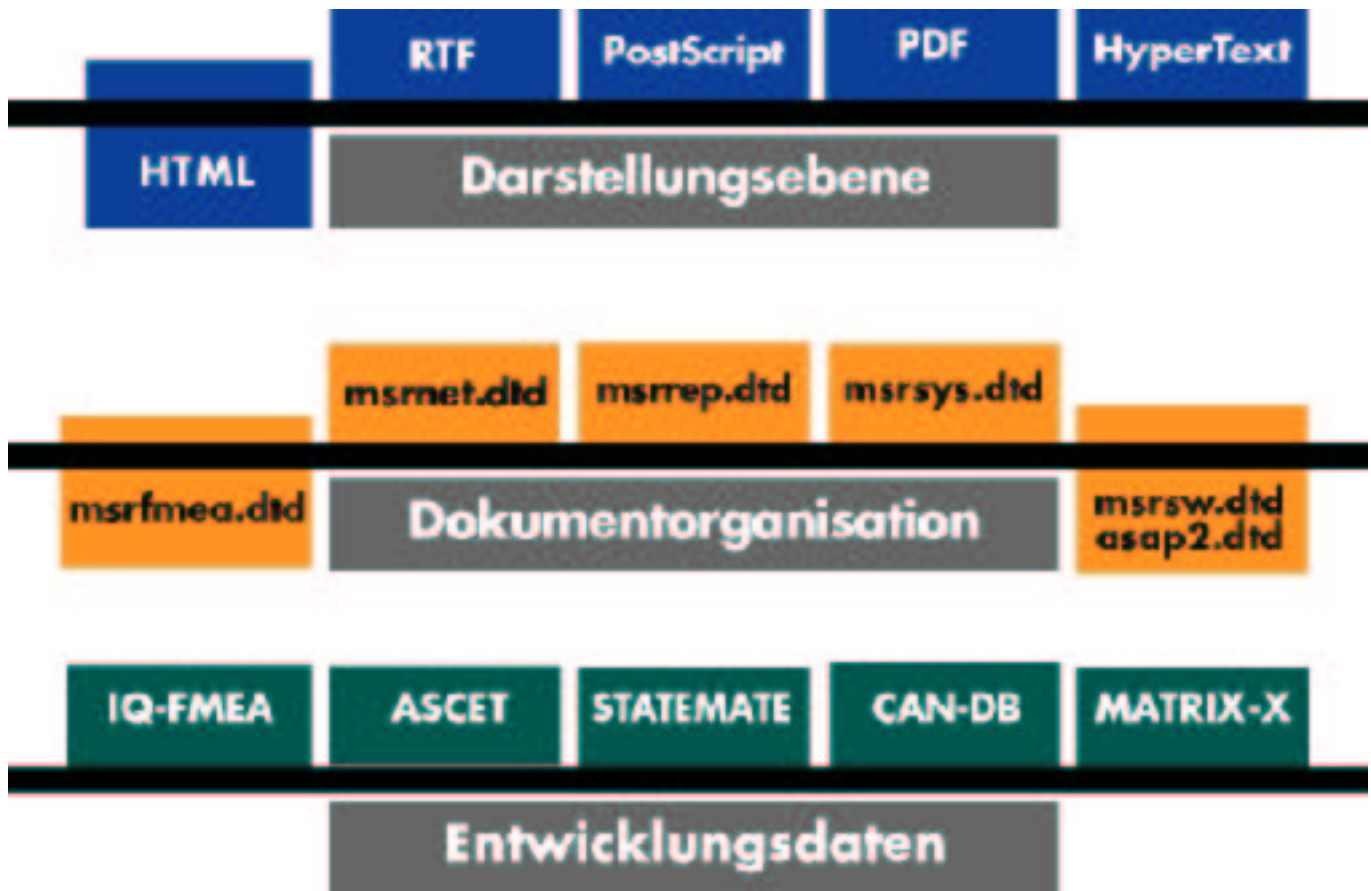


Abbildung 2: 3-Schichtenmodell

Die Darstellungsebene umfaßt die traditionelle Präsentationsweise von Information, etwa in Form eines gedruckten Dokumentes, bei dem die inhaltliche Strukturierung der Information mit Hilfe klassischer Layoutmittel und klassischer Navigationsmittel erschließbar wird.

Die Ebene der Entwicklungsdaten repräsentiert die Entwicklungsdaten in den Originalformaten der im Prozeß verwendeten Entwicklungswerkzeuge. Diese Dateien werden meist in EDMS (Engineering Data Management Systems) gehalten, welche keinen direkten Zugriff auf die verwalteten Inhalte besitzen.

Die Lücke zwischen Darstellungsebene und Entwicklungsdatenebene kann durch Einführung einer Zwischenschicht, der Ebene der Dokumentorganisation geschlossen werden. Diese Ebene muß auf einem standardisierten Format basieren, welches alle Informationen aufnehmen kann, die zum Erzeugen der gewünschten Präsentationen erforderlich sind.

Die Ebene der Dokumentorganisation ist die Domäne der SGML-Technologie. Die Trennung von Struktur, Inhalt und Layout ermöglicht eine vergleichsweise einfache Erzeugung bzw. Auswertung und gleichzeitig die Abbildung sehr komplexer Datenstrukturen.

3 Das MSR-Projekt

Das Projekt MSR (Manufacturer Supplier Relationship¹) ist ein Entwicklungsvorhaben mehrerer deutscher Automobilhersteller und Elektrik/Elektronik-Zulieferer mit der Zielsetzung, die Zusammenarbeit zwischen Hersteller und Zulieferer zu verbessern und in der Effizienz zu steigern. Es existieren mehrere Teilprojekte, welche unterschiedliche Themen behandeln.

Die Standardisierung von Schnittstellen zum Dokument- und Datenaustausch wurde im Teilprojekt MEDOC vorangetrieben.

Durch Untersuchung und Erprobung einer ganzen Reihe von Technologien zur Standardisierung und zum Austausch von Spezifikationen und Simulationsmodellen wurden die derzeit verfügbaren technischen Möglichkeiten ausgelotet. Als besonders tragfähig erwies sich SGML als Basis für die modulare, interaktive Zusammenarbeit bei Spezifikation und Dokumentation elektronischer Systeme. Die strukturellen Grundlagen für die komplette Beschreibung elektronischer Systeme wurden abgestimmt und in Pilotprojekten erfolgreich erprobt. Sie liegen als SGML-Dokument-Typ-Definitionen allgemein verfügbar vor.

Das Teilprojekt MEDOC hat bis zum Jahr 1995 die folgenden Phasen durchlaufen:

- Sammlung von Anforderungen
- Datenmodellierung anhand von Prototypen
- Auswahl einer Austauschtechnologie (SGML)
- Festlegung der Basiskonzepte zur Anwendung dieser SGML-Technologie
- Erstellung erster spezifischer Standards auf Basis von SGML.

Seit 1996 werden die Arbeiten auf dieser Basis weiterverfolgt. Dabei werden zum einen die Standards thematisch erweitert (z.B. zur Abdeckung von FMEA-Daten) und zum anderen die Basiskonzepte zur Anwendung der SGML-Technologie weiterentwickelt (z.B. um Konzepte für dokumentübergreifende Hypertextreferenzen). Im Bereich der Dokumentformatierung wurde ein gemeinsames Werkzeug entwickelt. Weitere Werkzeugimplementierungen werden in den beteiligten Firmen und von Drittanbietern auf Basis marktgängiger SGML-Werkzeuge (Editoren, Datenbanken, Konverter, ...) durchgeführt. Erste Kopplungen zu CAD-Werkzeugen sind ebenfalls am Markt verfügbar.

1

4 Was ist XML/SGML

Bei SGML handelt es sich um eine abstrakte Sprache zur Beschreibung von Dokumentstrukturen und Dokumentinhalten.

Dabei wird das Dokument durch SGML nicht in seinem Aussehen (Layout) beschrieben. Diese Art der Beschreibung ist von Vorteil, sobald das Layout unbedeutend ist, oder für alle Autoren gleich definiert sein sollte.

Ein sinnvoller Einsatz von SGML ist bei technischen Dokumenten gegeben. Technische Dokumente sind durch Langlebigkeit und durch strukturierte Informationen gekennzeichnet. Beispiele dafür sind Normen, Stücklisten, Konstruktionshandbücher, Lastenhefte, Pflichtenhefte sowie Fertigungs- und Bedienungsanweisungen.

4.1 Entwicklungsgeschichte

SGML besteht seit 1986 als internationaler ISO-Standard *8879 Information Processing Text and office systems Standard Generalized Markup Language (SGML)*, dessen Potential erst Anfang der 90er Jahre von einer größeren Anwendergruppe erkannt wurde. Die Grundidee stammt von dem IBM-Mitarbeiter Dr. CH. F. Goldfarb. Er entwickelte 1969 die Document Composition Facility General Markup Language (DCF GML), eine Auszeichnungssprache zur Batch-Formatierung von Dokumenten. DCF GML ist eine Makrosprache für ein Formatiersystem und daher mit Formaten wie TeX bzw. LaTeX vergleichbar.

DCF GML wurde dann in dem Standardisierungsgremium ISO/IEC JTC1/SC18/WG8 zu SGML verallgemeinert und um die Möglichkeit zur Festlegung einer Grammatik für die jeweilige Dokumentstruktur erweitert. Dadurch unterscheidet sich SGML von jeder anderen Auszeichnungssprache für Dokumente.

Das World Wide Web Consortium (W3C) hat im Februar 1998 die Entscheidung getroffen, eine Untermenge von SGML als zukünftige Sprache des World Wide Web einzuführen (unter dem Titel XML - eXtensible Markup Language). Dies wird weitreichende Folgen für die Verbreitung dieser Technologie als Basis für den zukünftigen Daten- und Dokumentaustausch in Computer- und Telekommunikationsanwendungen haben. Initiativen wie MSR MEDOC finden deshalb derzeit viele Nachahmer und bestehende Datenaustausch-Initiativen, die bisher auf anderen Technologien beruhen, stellen den Datenaustausch auf diese Technologie um.

4.2 Anwendungsbereiche

Es folgen einige Beispiele in denen SGML bereits eingesetzt wird.

- Luftfahrt
- Technische Dokumentation der Fluggeräte (ATA100-DTD)
- Automobilindustrie
- Wartungsanweisungen für Werkstätten (J2008-DTD)
- MSR-Konsortium: Lasten- und Pflichtenheftbeschreibungen
- Normungsinstitute bzw. Normungsabteilungen in der Industrie
- Z.B. DIN Normen (DINNORM-DTD)
- U.S. Verteidigungsministerium DoD²

- Sämtliche militärische Projekte wie z.B. die Dokumentation einer Fregatte (CALSDTD)³
- Verlagswesen
- Wissenschaftliche Veröffentlichungen
- Wörterbücher
- Computerindustrie
- Hypertext-Handbücher, Online Handbücher (OSF-DTD)
- Elektronikindustrie (CFI: PCIS-DTD)
- Elektronische Datenbücher für ICs und Standardbauteile
- World Wide Web WWW (HTML-DTD),
- Internet im Handy mit WAP mit WML DTD.
- E-Commerce-Anwendungen (EDIFACT-Ersatz)

4.3 L-Grundbegriffe

Ein Dokument besteht prinzipiell aus 3 Grundelementen:

Inhalt Dokumentendaten: zu vermittelnde Information in Schrift, Bild, etc.

Struktur Logischer Aufbau bzw. Gliederung des Dokumentes

Layout Darstellung der Information auf Papier, Bildschirm, etc.

Die einzelnen Teile sind zur Verdeutlichung noch einmal graphisch dargestellt.

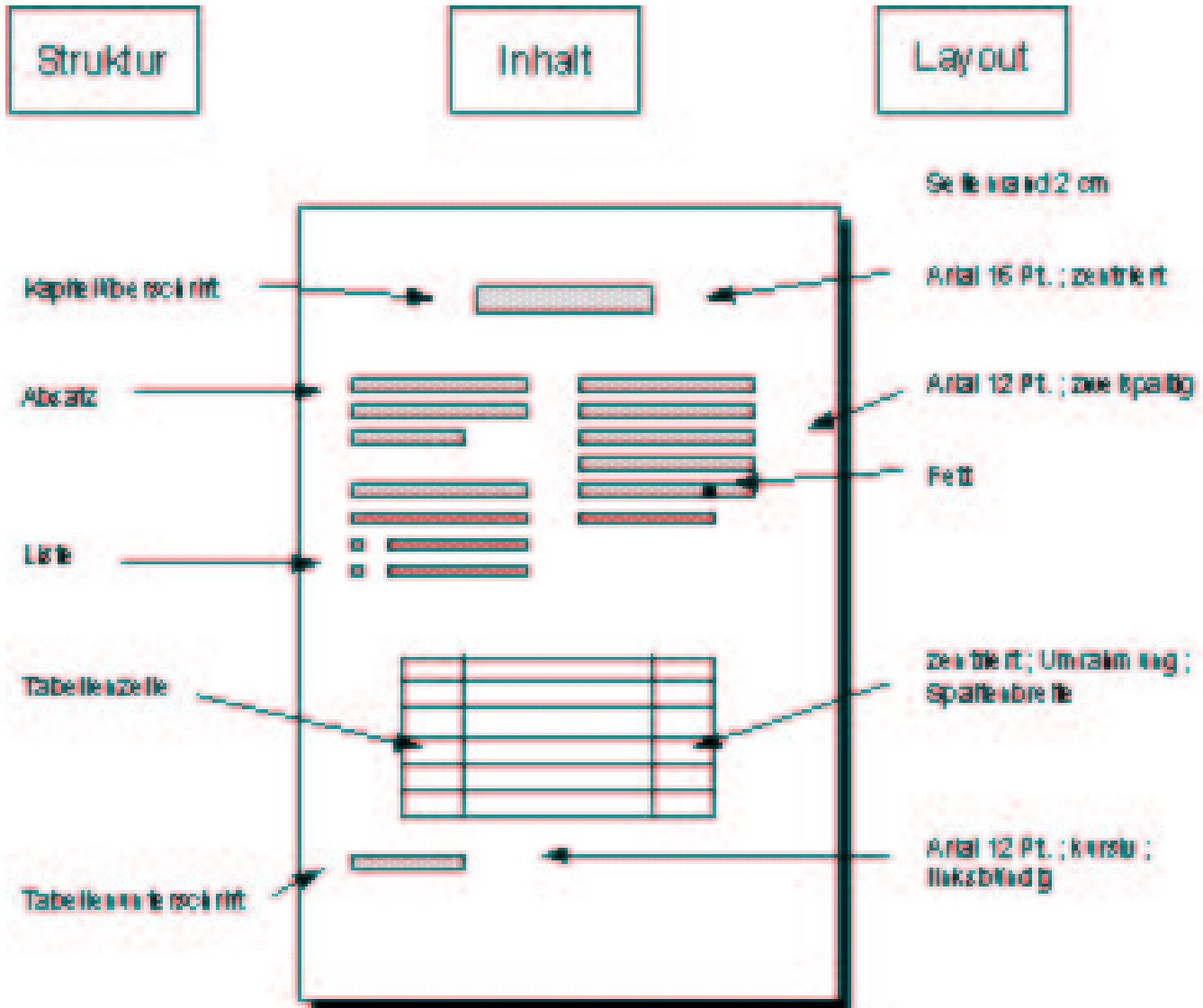


Abbildung 3: Inhalt, Struktur und Layout eines Dokumentes

Mit SGML werden die Dokumentobjekte und der logische Aufbau von Dokumenten einheitlich festgelegt. Damit kann die strukturelle Information von Dokumenten elektronisch übertragen und bearbeitet werden. Das visuelle Erscheinungsbild und damit das Layout wird allerdings von SGML explizit **nicht** abgedeckt. Die Standardisierung des Layouts ist jedoch der zweite Schritt, um ein Dokument vollständig elektronisch in einer systemneutralen Form verfügbar zu haben.

Die möglichen Strukturen innerhalb einer Dokumentenfamilie werden in der **Dokumenttypdefinition**, **DTD**, festgelegt.

Der Autor erstellt mit Hilfe eines SGML-Editors ein SGML-Dokument oder eine SGML-Quelldatei, die neben dem Inhalt auch die Strukturinformation in Form von Markup (engl. für Markierung) enthält. Das SGML-Dokument ist konform zu einer bestimmten DTD. Da zu einer DTD beliebig viele konforme Dokumente gehören können, spricht man von einer sog. **Dokumentinstanz**.

Ebenso wie die Strukturdefinition muß dem SGML-Editor auch die Layoutdefinition von außen zugeführt werden, damit die Bildschirmformatierung erzeugt werden kann. Bei der Layoutdefinition

gibt es zur Zeit die Standards XSL, CSS, DSSSL⁴ (sprich: diss"l) und FOSI⁵. Ziel der Standards ist eine allgemeine Layoutbeschreibung für alle möglichen SGML-Dokumente.

Ein SGML-Parser überprüft, ob die SGML-Syntax der DTD und der Instanz korrekt ist und wichtiger noch ob die Instanz konform zur DTD ist. Ebenso wird dabei die Vollständigkeit des Dokumentes überprüft. In allen SGML-Werkzeugen ist ein solcher SGML-Parser vorhanden.

Eine SGML Instanz kann nun auf verschiedene Arten weiterverarbeitet werden. Mit einem **Formatierer** kann das Dokument zu Papier gebracht werden, mit einem **Viewer** ist die elektronische Darstellung möglich und mit einem **Konverter** ist eine Konvertierung zum Beispiel in ein CAD-Format möglich.

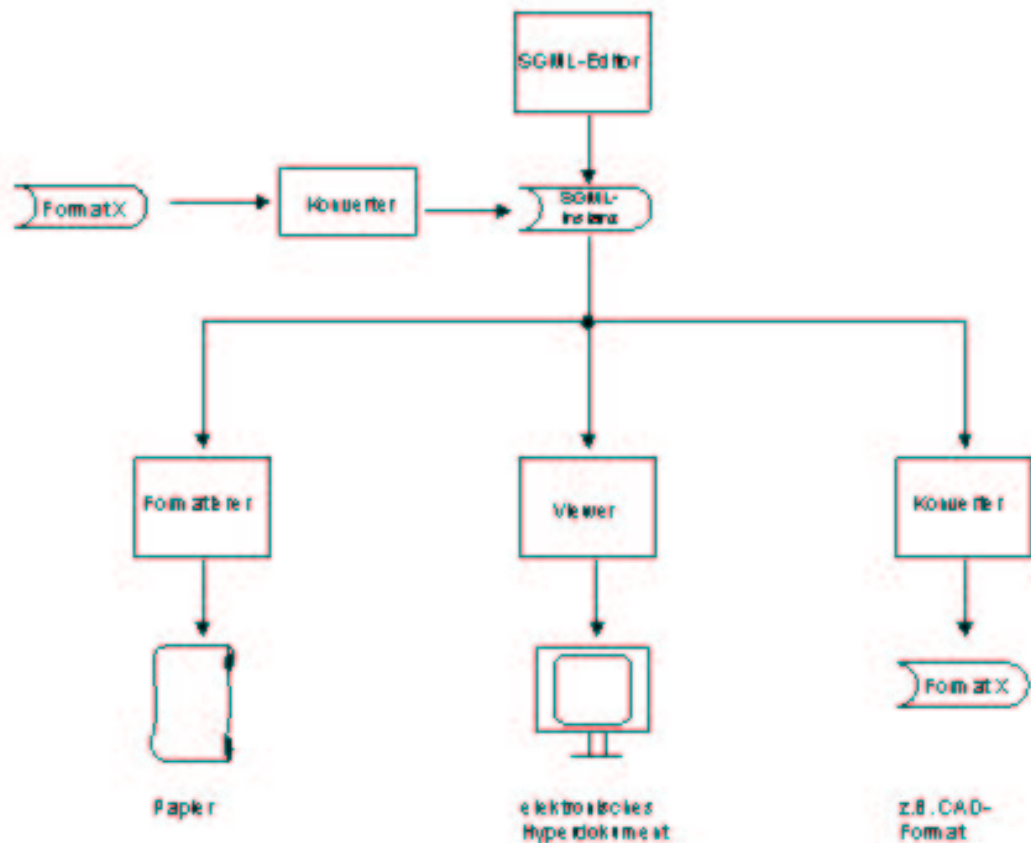


Abbildung 4: Generelles SGML-Szenario

Aus dieser Grafik wird deutlich, daß im Gegensatz zu anderen DTP-Anwendungen eine Trennung der verschiedenen Prozesse stattfindet.

4
5

4.4 Bestandteile eines SGML-Dokumentes

Ein SGML-Dokument setzt sich aus drei Bestandteilen zusammen:

- der SGML-Deklaration
- der DTD
- der SGML-Instanz

Prinzipiell verwendet SGML den 7-Bit ASCII-Code⁶. Es sind aber Referenzen auf Dateien im Nicht-SGML"-Format wie z.B. Grafiken möglich.

4.4.1 Die SGML-Deklaration

An dieser Stelle soll nicht weiter auf die SGML-Deklaration eingegangen werden, da sie hauptsächlich für den DTD-Entwickler von Bedeutung ist.

4.4.2 Die DTD eine baumartige Dokumentstruktur-Definition

Bei der DTD handelt es sich um eine baumartige⁷ Struktur, in der die Reihenfolge und die hierarchischen Abhängigkeiten der für das Dokument relevanten Textteile (**Elemente**) festgelegt und damit die Tags (engl. für Markierungen) für die später zu erstellenden Dokumente definiert ist. Den Elementen können in der DTD **Attribute** zugeordnet werden, welche die Eigenschaften der Elemente repräsentieren⁸ und die auch die Querverweise in SGML realisieren.

Elemente und ihre Attribute bilden somit die grundlegenden Mechanismen dieses Standards.

Bevor Dokumente mit SGML bearbeitet werden können, müssen ihre logischen Baumstrukturen in einer DTD festgelegt werden. Eine solche logische Struktur kann z.B. sein, daß ein Dokument immer mit einem Titel beginnt, der von einer nicht notwendigen Autorenangabe und diese wiederum von mehreren Kapiteln gefolgt wird. Außerdem besteht die Struktur eines Kapitels aus einer Überschrift, gefolgt von Absätzen, Aufzählungen oder Unterkapiteln.

Es ist also zwingend erforderlich, die Zieldokumente gründlich zu analysieren, um die Dokumentstrukturen zu erkennen und in die DTD umzusetzen. Es ist deshalb von Vorteil, die Autoren bereits beim Festlegen der DTD hinzuzuziehen. Generell ist eine DTD-Entwicklung eine gemeinsame Aufgabe von SGML-Experten und Autoren.

Folgenden wesentliche Eigenschaften werden in einer DTD definiert:

- Die Namen der Elemente, die erlaubt sind.
- Wie oft ein Element erscheinen darf.
- Die Reihenfolge, in der die Elemente erscheinen müssen.
- Den Inhalt eines Elementes, z.B. ob innerhalb eines Elementes ein weiteres Element auftauchen darf oder nicht.
- Die Namen der Entities, die verwendet werden können.
- Attribute

6

7

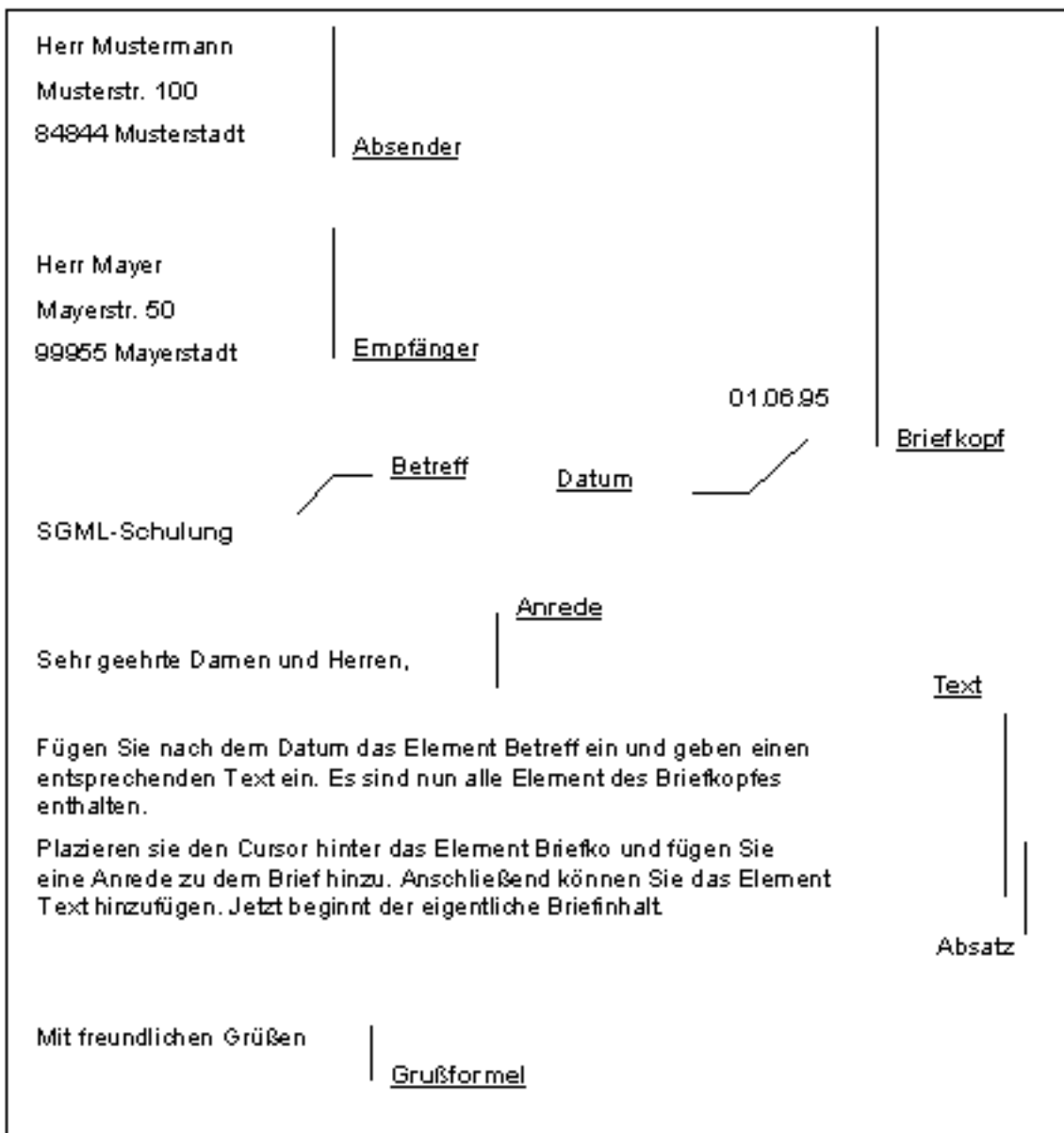
8

4.4.3 Die SGML-Instanz

Die SGML-Instanz stellt das eigentliche Dokument dar. Generelle Elemente einer solchen Instanz sind:

- **Elemente und Attribute**
- **Text**
- **Entities**

Als Beispiel ist hier ein einfacher Brief angeführt, der zunächst als normaler Text dargestellt ist, wie folgende Abbildung zeigt.



Brief im Layoutformat

Abbildung 5: Brief im Layoutformat

Ein Brief besteht aus mehreren Elementen. Prinzipiell gibt es natürlich viele Möglichkeiten, die Struktur eines Briefes festzulegen.

Die hier zugrunde gelegte Struktur besteht aus den vier Hauptelementen Briefkopf, Anrede, Text und Grußformel (in der Abbildung fett gedruckt). Diese Elemente haben zum Teil wieder Unterelemente. Das Element Briefkopf besteht aus den Unterelementen Absender, Anschrift, Datum und Betreff. Das Unterelement Text wiederum kann aus einem oder mehreren Absätzen bestehen.

Der Aufbau einer solchen Struktur läßt sich relativ anschaulich in einem Strukturdiagramm darstellen, wie die folgende Abbildung zeigt⁹.

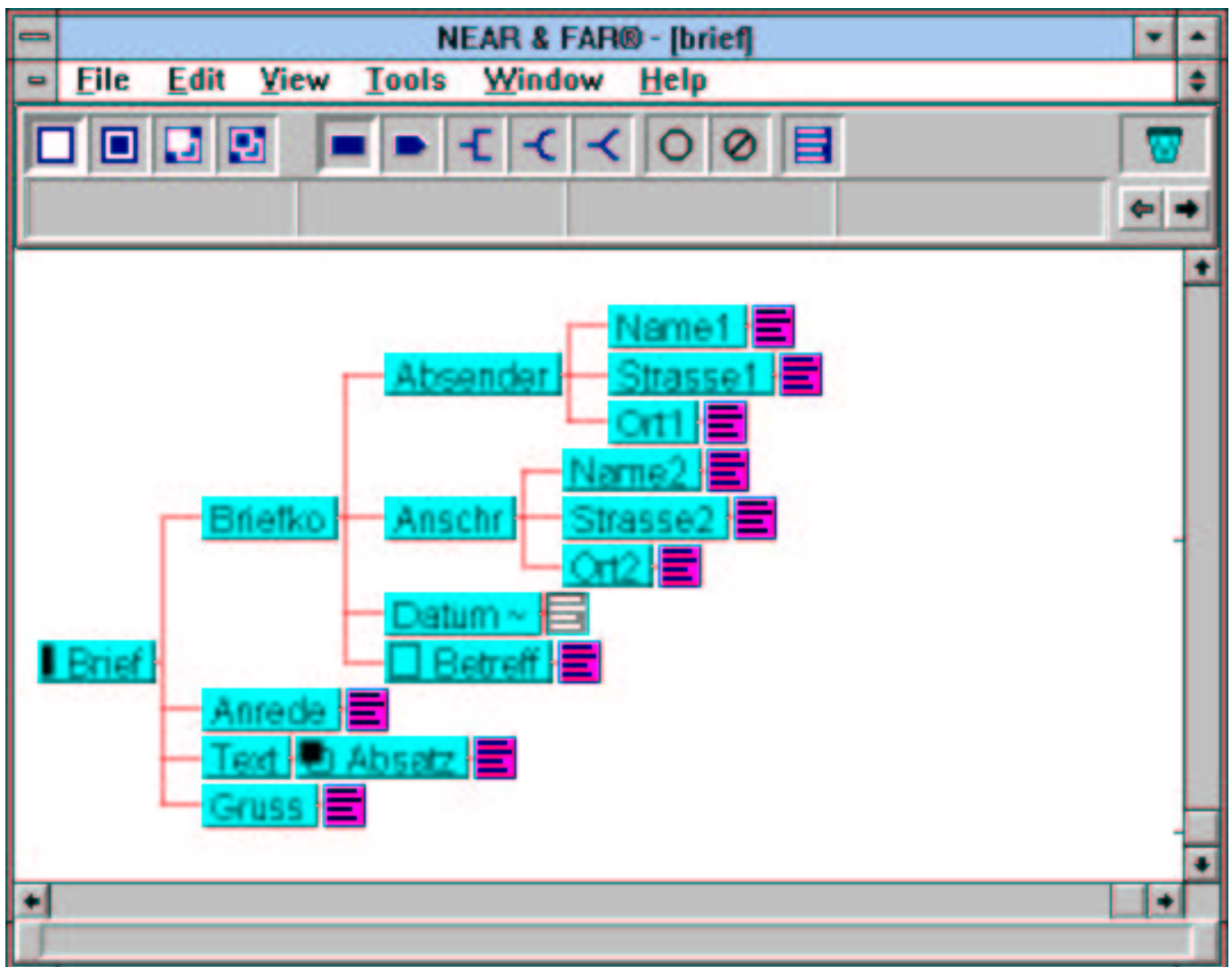


Abbildung 6: Strukturdiagramm des Briefes

Die nächste Abbildung zeigt den gleichen Brief im SGML-Format.

Brief im SGML-Format

⁹

Die erste Zeile beinhaltet den Namen der verwendeten DTD (hier: BRIEF.DTD). Die einzelnen zuvor beschriebenen Elemente sind jetzt in sog. Markierungen eingeschlossen, wobei folgende Darstellung gewählt wird:

Start-Markierung (start-tag) <Element>

Ende-Markierung (end-tag) </Element>

Zwischen der Start- und Ende-Markierung ist der jeweilige Elementinhalt (Text, Nichts oder weitere Elemente) eingeschlossen. Da in SGML nur der ASCII-Zeichensatz verwendet wird, werden Sonderzeichen als Character-Entities, d.h. als genormte SGML-Zeichenstrings (z.B. ü für "ü") dargestellt.

Die folgende Grafik zeigt das selbe Dokument in einem sog. SGML-Editor. Als Editor wurde hier der "Author/Editor" von SoftQuad verwendet. Die Markierungen sind hervorgehoben und außerdem mit einer Layoutdefinition versehen, d.h. die Darstellung auf dem Bildschirm kann, muß jedoch nicht dem späteren Ausdruck auf Papier entsprechen.

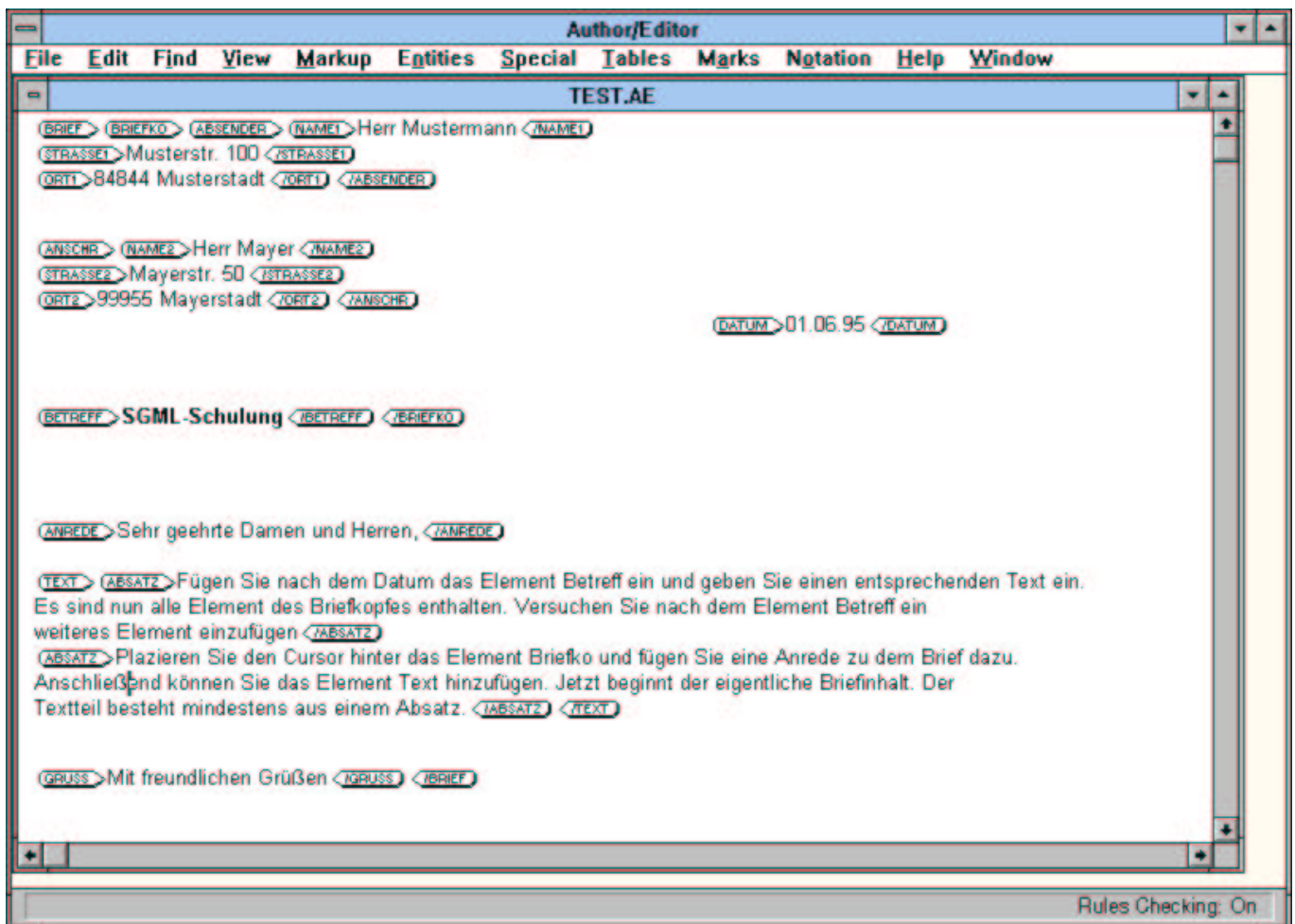


Abbildung 7: Darstellung eines Dokumentes mit einem SGML-Editor

5 MSR-Standards

Der erste Schritt zur Anwendung der SGML-Technologie war die Definition von einheitlichen Grundsätzen und Regeln zum Umgang mit dieser Technologie (MSR MEDOC Applikationsprofil). Ziel dabei ist die Interoperabilität der einzelnen entstehenden MSR-Standards sowie die Unterstützung der Wiederverwendung von Software, die die MSR-Standards implementiert.

5.1 Das SGML Applikationsprofil von MSR MEDOC

Die MEDOC-Datenmodelle werden in SGML implementiert. Dies geschieht bei SGML durch die Definition sogenannter DTDs (Document Type Definitions). Alle MSR DTDs haben die folgenden, im Applikationsprofil festgelegten, einheitlichen Prinzipien:

Hyperlink-Modell

Das MSR Hyperlink-Modell unterstützt diverse Möglichkeiten, Hyperlinks innerhalb eines Dokuments oder auch zwischen unterschiedlichen Dokumenten zu definieren. Damit wird es ermöglicht, die Gesamtheit der verschiedenen Dokumente zu einem Steuergerät (System- und HW-Lastenheft, Softwaredokumentation, Vernetzungsbeschreibung, etc.) als eine große, vernetzte "Steuergeräte-Datenbank" aufzufassen.

Basiselemente

In allen DTDs werden die selben Basiselemente verwendet. Diese Basiselemente umfassen u. a. alle Elemente zur reinen Textdarstellung (z.B. zahlreiche Listenformen) aber darüber hinaus auch einheitliche Modelle wie z.B. die Darstellung von Parametern oder die Behandlung von physikalischen Einheiten.

Konfigurierbarkeit

Die zahlreichen Konfigurierungsmöglichkeiten in den DTDs dienen u.a dazu, sich an unterschiedliche Prozeßschritte oder Werkzeugumgebungen anpassen zu können.

Klassifikationsmechanismen

Die DTDs beinhalten ein einheitliches Klassifikationskonzept, das über Namensweiterungen der Datenelemente abgebildet ist.

Administrative Daten

Die administrativen Daten umfassen Versions-, Änderungs-, u.ä - Informationen und erlauben so die Kopplung der MSR-Daten an Verwaltungssysteme (Dokumentenmanagement-Systeme, Produktdatenmanagement-Systeme o.ä.).

Konstruktionsmechanismen für die DTDs

Alle DTDs werden nach festgelegten Konstruktionsmechanismen gebildet (z.B. Namenskonventionen).

5.2 Die einzelnen MSR-Standards

Im folgenden sind die derzeit festgelegten MSR-Standards aufgelistet und erläutert.

System (MSRSYS DTD)

Die MSRSYS DTD (früher MSRDOC) dient zur Spezifikation und Dokumentation kompletter Elektrik/Elektronik-Systeme. Sie erlaubt u.a. die folgenden Beschreibungen:

- Projektdaten
- Bauteile und ihr hierarchischer Aufbau



- Verhalten
- Architektur (Signale, Schnittstellen, Verbindungen)
- Elektrische Eigenschaften
- Mechanische Eigenschaften
- Optische und akustische Eigenschaften
- Umweltverträglichkeit

Die DTD wurde in verschiedenen Autorenumgebungen implementiert und befindet sich in zahlreichen Projekten im Einsatz.

Software (MSRSW DTD)

Die MSRSW DTD dient zur Spezifikation und Dokumentation von Steuergerätesoftware und umfaßt die folgenden Bereiche:

- Datenlexikon
- Funktionale Spezifikation
- Kenngrößen

Erste Tool-Kopplungen für diese DTD sind am Markt verfügbar.

Network (MSRNET DTD)

Die MSRNET DTD dient zur Spezifikation und Dokumentation der Fahrzeugvernetzung (mit dem aktuellen Fokus auf CAN-Netzwerke). Sie erlaubt u.a. die Beschreibung der Netzwerktopologie sowie der auf dem Netz ausgetauschten Signale und Botschaften. Erste Implementierungen der MSRNET DTD sind am Markt verfügbar und Pilotanwendungen wurden gestartet.

Diagnose

Im Februar 98 wurde eine Arbeitsgruppe gegründet, die eine DTD für Diagnose-Lastenhefte erstellt. Diese DTD hat viele Anknüpfungspunkte zu anderen DTDs und damit die Aufgabe, eine integrierte Darstellung der Diagnosegesichtspunkte zu ermöglichen. Ein weiterer Anknüpfungspunkt bildet die im ASAM/ASAP-Projekt definierte SGML DTD zur Beschreibung der Kommunikation von Steuergeräten mit Off-Board-Diagnose- und Testsystemen. Sowohl zur Interoperabilität dieser DTDs als auch zur inhaltlichen Aufteilung der Arbeiten finden entsprechende Abstimmungen zwischen MSR MEDOC und ASAM/ASAP statt.

Report (MSRREP DTD)

Dies ist eine allgemein einsetzbare DTD, die nur zu Publikationszwecken und zum Dokumentaustausch, nicht aber zum Datenaustausch eingesetzt wird. Sie findet derzeit Anwendung für sämtliche Dokumentationen, die in den MSR MEDOC Arbeitsgruppen erstellt werden, sowie auch für Anwendungsgebiete, die noch nicht mit einer MSR DTD abgedeckt sind. Neben einem allgemeinen Dokumentmodell mit Kapiteln, Paragraphen etc. unterstützt die MSRREP DTD auch eine strukturierte Alternativenbehandlung (z.B. für den Änderungs- und Versionsplanungsprozeß).

FMEA (MSRFMEA DTD)

Die Failure Mode and Effect Analysis (FMEA) DTD ist eine Struktur welche eine Methode für die systematische Vorgehensweise zur Identifikation von Fehlerquellen und Fehlerwirkungsketten beschreibt.

Dieses Verfahren ist für verschiedene Bereiche (Geräte, Systeme, Software, aber auch für Abläufe und Prozesse) anwendbar.

6 Die MSR FMEA DTD

6.1 Ziele der MSRFMEA.DTD

Die FMEA ist ein Baustein in der System-Entwicklung. Die DTD wurde entwickelt, um die in spezifischen *FMEA-Werkzeugen* erfaßten Daten in anderen Entwicklungsschritten verwenden zu können. Dazu gehören beispielsweise:

- Aufbau von Standards (Katalogen, Bibliotheken)
- Terminverfolgung
- FMEA-übergreifende Auswertungen
- Langfristige Datenhaltung (Archivierung)
- Generierung von Prüfabläufen aus FMEA
- Datenaustausch mit anderen Systemen (z.B. CAD, CASE, QFD)

Die Struktur sollte alle Daten welche in diesen Prozessen und in den FMEA-Tools verwendet werden abbilden.

Folgende Grafik verdeutlicht die in der DTD abgebildete System-FMEA-Methodik:

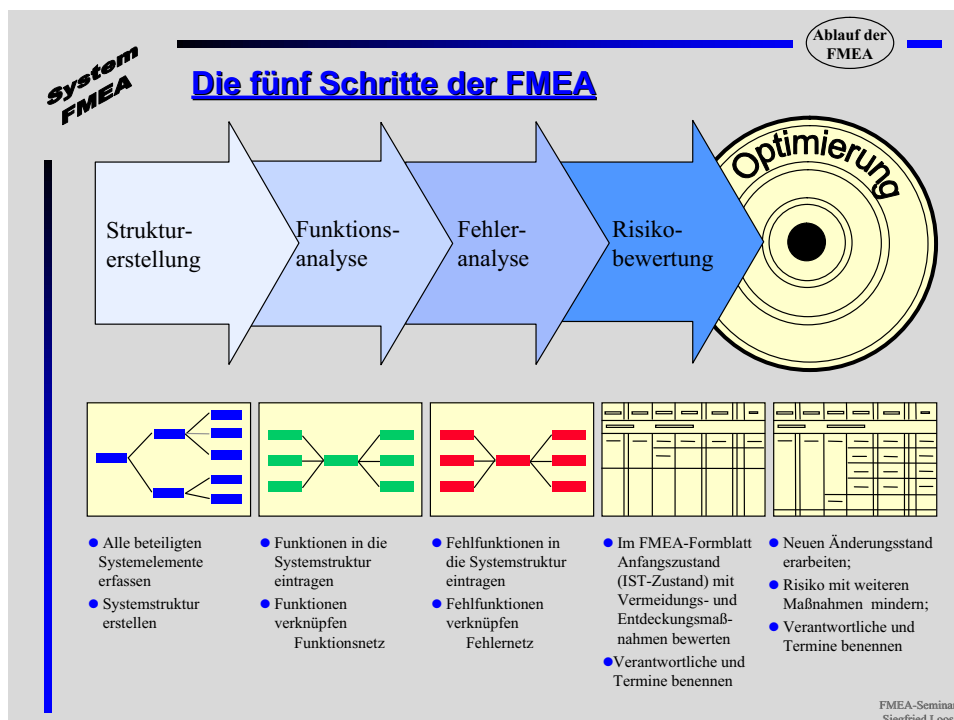


Abbildung 8: Erstellung einer System-FMEA

6.2 Prozeßmodelle für die DTD

Die folgenden Use-Cases wurden beim Entwurf der DTD betrachtet.

6.2.1 Hinzuladen von Maßnahmen bzw. Funktionskatalogen

In eine bestehende oder neu aufzubauende FMEA sollen Maßnahmen bzw. Funktionskataloge hinzugeladen werden:

- Die DTD kann teilweise bedatet werden, d.h. es ist möglich, Maßnahmen ohne Strukturelemente zu transportieren. Daher enthält die DTD auch keine Rückverzeigerung (z.B. von Funktionen auf Strukturelemente).
- Für die Identifizierung der Objekte müssen IDs vergeben werden. Im Konfliktfall kann ein lesendes System neue IDs vergeben und die vorgefundenen IDs unter Angabe des ursprünglichen Werkzeuges ablegen (`<fm-idtables>`)

6.2.2 Manipulation der FMEA mit anderen Werkzeugen

In diesem Szenario wird die FMEA exportiert, mit anderen Werkzeugen manipuliert, und wieder importiert. Beispiel für diese Anwendung ist das Hinzufügen weiterer Sprachvarianten durch ein externes Übersetzungsbüro.

- Die Objekt-ID kann zur Resynchronisation verwendet werden, darf daher nicht extern manipuliert werden. Es gibt keine Einschränkungen hinsichtlich Änderungen, solange IDs nicht betroffen sind. ID-Änderungen unterbrechen die Prozeßkette.
- Im Fall nicht strukturverändernder Manipulation sind die Auswirkungen bekannt, so daß der Ereignisraum begrenzt wird. Es brauchen z.B. keine Löschungen von Objekten behandelt zu werden.

6.2.3 FMEA-Erstellung auf Basis von Engineering-Tool Daten

In diesem Szenario wird in *FMEA-Tool* eine Datenanreicherung vorgenommen. Die Komponenten/Funktionsstruktur kommt aus dem Entwicklungswerkzeug z.B. durch Auswertung von Instanzen der *MSRSYS.DTD*.

Hierbei ist problematisch, wenn im Engineering-Tool im Rahmen der Weiterentwicklung eine Strukturveränderung stattfindet.

Es erfolgt die Synchronisation mit IDs, wobei folgende Fälle zu betrachten sind:

neue ID Objekt ist neu entstanden

fehlende ID Objekt ist gelöscht

ID bereits vorhanden Es muß geprüft werden, ob das Objekt sich geändert hat.

Dieser Fall kann ggf. durch Synchronisation mit Änderungsdatum unterstützt werden. Dafür wird das Attribut `[t]` vorgesehen, welches Zeitstempel aufnehmen kann.

6.2.4 Wiederherstellen einer FMEA aus der SGML-Instanz

In diesem Szenario wird die FMEA nach SGML exportiert und zu einem späteren Zeitpunkt wieder hergestellt. Es ist allerdings nicht notwendig die im FMEA-Tool abgelegten Verwaltungsdaten (z.B. Historie) wieder herzustellen. Das Wiedereinlesen einer SGML-Datei entspricht einer "automatisierten Neueingabe".

6.2.5 Modularisierung

Modularisierung von FMEAs ist methodisch noch in der Diskussion und wird daher beim ersten Schritt nicht priorisiert betrachtet. Schnittstellendefinition (`<fm-interface>`) werden jedoch aufgenommen, um Erfahrungen gewinnen zu können.

6.3 Grobstruktur der DTD

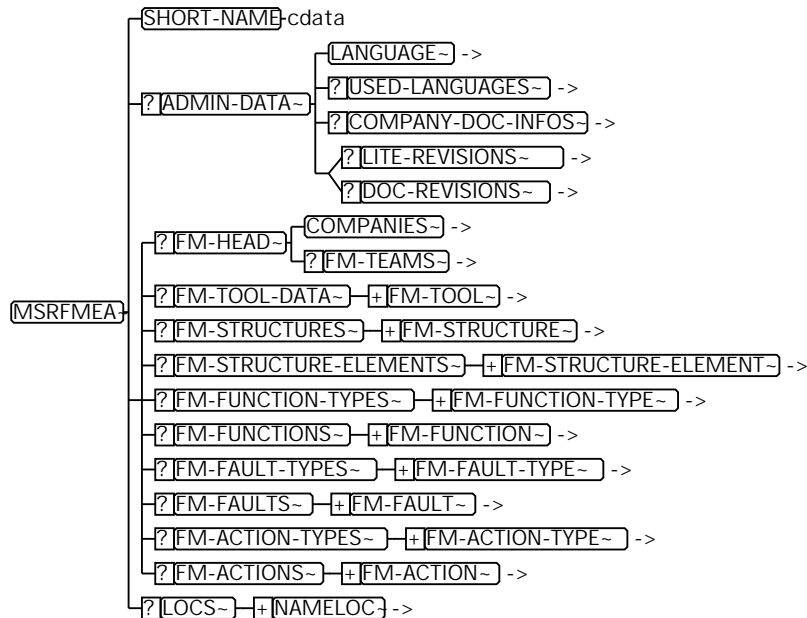


Abbildung 9: Grobstruktur MSRFMEA DTD V1.5.0

Die Bedeutung der Elemente auf oberster Ebene ist:

v short-name

Dieser dient zur Identifikation der SGML-Datei, insbesondere bei dokumentübergreifenden Verweisen. Es kann z.B. die Kurzbezeichnung eines Produktes (z.B. LLR) verwendet werden. Ein Exportfilter könnte hier auch den Basisnamen der Datei verwenden.

v admin-data

Dies dient zur Aufnahme von Verwaltungsinformationen.

v fm-head

Dies enthält die am Projekt beteiligten Firmen **companies** sowie Informationen zu den beteiligten FMEA-Teams **fm-teams**.

v fm-tool-data

Dient zur Aufnahme werkzeugspezifischer Informationen, insbesondere hinsichtlich der ID-Verwaltung (Unterstützung von Tool-Besonderheiten). In v **misc-data** innerhalb von **fm-tool** kann ein Werkzeug private Daten hinterlegen, z.B. um einen optimierten Wiederimport zu gewährleisten.

v fm-structures

Hier werden Strukturen des Systems bzw. des Prozesses hinterlegt.

v fm-structure-root

Über dieses Element findet man den Einstieg in den Strukturbaum.

V fm-structure-elements

Hier werden die Einzelkomponenten des Systems beschrieben. An den Strukturelementen hängen die entsprechenden Funktionen x

fm-se-functions

An den so referenzierten Funktionen hängen wiederum die Fehlfunktionen.

x fm-functions

Dies ist der Pool sämtlicher im System vorhandener Funktionen (strukturübergreifend). Diese werden dann aus den Strukturelementen heraus referenziert.

x

fm-faults

Dies ist der Pool sämtlicher Fehler, die im System auftreten können. Diese werden aus den Funktionen heraus referenziert. Die DTD erlaubt es, daß ein Fehler in mehreren Funktionen referenziert wird. Es wird aber empfohlen, dies nicht zu nutzen, da keine verwendungsspezifischen Eigenschaften angegeben werden können.

x

fm-actions

Dies ist der Pool sämtlicher Maßnahmen, die im System auftreten können. Diese werden aus den Fehlern heraus referenziert.

Die DTD erlaubt es, daß eine Maßnahme in mehreren Fehlern referenziert wird. Es wird aber empfohlen, dies nicht zu nutzen, da keine verwendungsspezifischen Eigenschaften angegeben werden können.

x

fm-function-types

Dies enthält die im System vorhandenen Funktionstypen.

x

fm-fault-types

Dies enthält die im System vorhandenen Fehlfunktionstypen.

x

fm-action-types

Dies enthält die im System vorhandenen Maßnahmentypen

x

locs

Dies enthält die Elemente zur dokumentübergreifenden Referenzierung.

Die vollständige Dokumentation zur DTD ist in Zukunft unter /MSR98/ erhältlich.



7 Literaturverzeichnis

[Externes Dokument: MSR WG Home Page / URL: <http://www.msr-wg.de>]



8 Referent

Herbert Klein
XI-WORKS
Olgastraße 86
D-70180 Stuttgart
TEL.: +(49) 711/248398-0
FAX.: +(49) 711/248398-29
Email: herbert.klein@xi-works.de



Dokumentverwaltung

Tabelle : Beteiligte Personen

Name	Firma	
Herbert Klein	MSR MEDOC	Telefon: +49 711 248398-10 Abteilung: XI-Works Herbert.Klein@xi-works.de
Roman Reimer, XI-Works	MSR MEDOC	Abteilung: XI-Works

Tabelle : Versionsübersicht

Version	Datum	Herausgeber	Status
1	1999-10-12	Herbert Klein	RD