

SGML für die Praxis

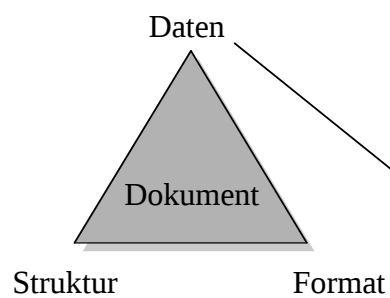
ISO 8879

Von W. Rieger

Springer

(I.7. EDV 10335)

Dokumente:



- Dokumente sind aufgezeichnete Informationen
- Multimediadokumente integrieren mehrere Formen von Informationen (Text, Bilder, Grafiken, Tabellen, Video, Musik, Sprache, Animationen, interaktive Programme, ...)

Dokumente: Tools

- Editoren: zur Texterstellung
 - ◆ VI, Emacs, ...
- Textverarbeitungsprogramme
 - ◆ Microsoft Word, WordPerfect, ...
- DTP-Programme
 - ◆ PageMaker, QuarkXPress
- WYSIWYG: What you see is what you get

Dokumente: Formate

- Seitenbeschreibungssprachen
 - ◆ PostScript, RTF, PDF, ...
- ISO: International Organization for Standardization
 - ◆ DIN: Deutsches Institut für Normung
 - ◆ ANSI: American National Standards Institute
 - ◆ BSI: British Standards Institution

Dokumente: SGML

- Standard Generalized Markup Language
- Lösung des Problems, Struktur und Daten eines Dokuments portabel und flexibel wiederzugeben
- Standardisiertes Dateiformat zum Austausch von Dokumenten
- ISO 8879 Standard zur Beschreibung von Dokumentstrukturen
- Deskriptive Sprache mit definierter Syntax zur Beschreibung von Dokumentstrukturen

Dokumentstruktur

- Ergibt sich aus der Abstraktion der Dokumentinstanz
- Die Bestandteile eines Dokuments eines Dokument-Typs (Dokumentklasse) werden mit tags markiert.
- Tag: `<Elementname>Inhalt</Elementname>`
- Sonderzeichen: `&Symbolname;`

Dokumentstruktur: Forderungen and die Markierung

- Abstraktion von Konstanten der Dokumentart
 - ◆ Bestandteile, die allen Instanzen angehören, gehören nicht zum Inhalt (z.B. Firmenlogo,...)
- Semantisch korrekte Markierung
- Vollständige Markierung
- Weitgehende Implizierung von Inhalten
 - ◆ Verweise, Inhaltsverzeichnisse, Index

Dokumentstruktur: Arten

- Hierarchisch
 - ◆ linear dargestellt, oder als Hypertext
- Linear
- Nichtlinear
 - ◆ Verweise, Fussnoten, Marginalien
- Bewegliche Bestandteile
 - ◆ können an beliebiger Stelle im Text auftreten
- Text versus Hypertext
 - ◆ Hoher lienearer Anteil
 - ◆ Hoher nichtlinearer Anteil

Dokumentstruktur: Beispiel

```
<geschäftsbrief>  
<adr>  
<firma>IfI  
<strasse>Winterthurerstrasse 190  
<plz>8050  
<stadt>Zürich  
</adr>  
<anrede>Sehr geehrte Herren  
...  
<gruss>Mit freundlichen Grüßen  
</geschäftsbrief>
```

SGML im Einsatz

- Einsatz sinnvoll wenn die Dokumente eine ausgeprägte Struktur haben
 - ◆ Adäquater als PostScript (Seitenbeschreibung)
 - ◆ Nicht zu regelmässig
(sonst Datenbank besser)

SGML im Einsatz: Neue Medien

- Online-Dokumente
 - ◆ reusable documents (Mehrfachverwendung)
- Database Publishing (SGML als Austauschformat)
- Electronic Publishing
- Hypertext
- Hypermedia

SGML im Einsatz: Einsatzbereiche

- Dokumentation und Manuale
 - ◆ Strukturelemente wie Name, Kurzbeschreibung, Synopsis, Beschreibung, Fehlerliste, „Siehe auch“
- Gebrauchsanweisungen und Reparaturanleitungen
 - ◆ DoD verwendet SGML als Standard für die Beschreibung militärischer Dokumentation
- Technische Texte, Wissenschaftliche Texte
 - ◆ Austauschbarkeit ist wichtig
- Formulare, Rechnungen, Bestellungen, ...

SGML im Einsatz: Einsatzbereiche

■ Gesetzestexte, Normen

- ◆ Gesetz gliedert sich in Artikel, Paragraphen, Absätze, Sätze, Nummern, Buchstaben
- ◆ Gründung von SGML in Anwaltsbüro durch Goldfarb, Mosher, Lorie (GML)
- ◆ DIN verwendet SGML zur Beschreibung von Normtexten

■ Lexika, Wörterbücher

■ Kataloge

■ Reihen und Periodika

- ◆ nur mit regelmässigem , strukturierten Inhalt

SGML im Einsatz: Ungeeignet für

■ Chaotische Struktur

- ◆ Layout dominiert Inhalt ➡ DTP Programme

■ Flache Struktur

■ Fehlende Struktur

- ◆ Romane

■ Illustrierte

Muster und Modelle: **Klassen und DTD**

- Dokument-Type ist Klasse
- Dokument ist Instanz einer Klasse
- DTD: Dokument-Typ-Definition
 - ◆ Bestimmt Struktur einer Klasse
 - ◆ Wird gegeben durch Grammatik oder Strukturmuster
 - ◆ Bspl: Geschäftsbrief, Bestellung, ...

Muster und Modelle: **DTD Notation**

- Primitive Muster
 - ◆ (#PCDATA) Parsable Character Data
 - ◆ #: kein Element, sondern spezieller Bezeichner
 - ◆ (mail)
- Konnektoren
 - ◆ Sequenz: (anfang, mitte, schluss)
 - ◆ Repetition: (akt+) oder (akt*)
 - ◆ Option: (anfang?, mitte+, schluss?)
 - ◆ Beliebige Reihenfolge: (a & b & c)
 - ◆ Alternative: (schwarz | weiss)

Muster und Modelle: DTD Notation

■ Gruppen

- ◆ (Absatz, (Absatz | Liste | Tabelle)*)

■ Mehrdeutigkeit

- ◆ Strukturbeschreibung muss eindeutig sein
 - ✦ (A+, B?, A+) ist z. B. nicht eindeutig
A A A

■ SGML Parser

- ◆ Liest DTD
- ◆ Liest Dokument und überprüft Struktur
- ◆ Meldet Fehler

SGML Dokumente

■ SGML Dokument besteht aus

- ◆ SGML Deklaration (fehlt meistens, Voreinstellung)
- ◆ Prolog (enthält Namen des DT und DTD)
- ◆ Dokumentinstanz

■ DTD besteht aus

- ◆ Externen Bezeichner
- ◆ Inhalt des DTD-Subsets (Elementdeklarationen)

SGML Dokumente: Prolog Syntax

■ Prolog Syntax

◆ <!DOCTYPE

DTD-Name

externer-DTD-Bezeichner ?

([DTD-Subset-Deklaration]) ? >

- Inhalt des Subsets ergänzt und modifiziert die Deklarationen einer eventuell vorhandenen externen DTD

SGML Dokumente: Elementdeklaration

■ Die Syntax einer Elementdeklaration

◆ <!Elementname

Start-Minimierung

Ende-Minimierung

Inhaltsmodell

Exklusionen

Inklusionen>

SGML Dokumente: Einfaches Beispiel

■ Beispiel eines einfachen SGML Dokumentes

- ◆ `<!DOCTYPE einfach [
 <!ELEMENT einfach - - (#PCDATA) >
]>
 <einfach>Ein sehr einfaches Dokument.</einfach>`
- ◆ Es fehlt die SGML Deklaration

SGML Dok.: Hierarchische Struktur

■ Beispiel einer Text Struktur

- ◆ `<!DOCTYPE text [
 <!ELEMENT text - - (kopf, (absatz | liste)+) >
 <!ELEMENT kopf - - (#PCDATA)
 <!ELEMENT absatz - - (#PCDATA)
 <!ELEMENT liste - - (lp)*
 <!ELEMENT lp - - (#PCDATA)
]>`
- ◆ `<text>
 <kopf>Kopf des Textes</kopf>
 <absatz> 1. Absatz</absatz>
 <liste> <lp>Listenelement 1</lp> </liste>
 <absatz> 2. Absatz</absatz>
<text>`

SGML Dok.: Minimierung von Markierungen

- Man möchte die Anzahl der Start- und Endmarkierungen möglichst reduzieren um das Dokument übersichtlich zu gestalten.

- ◆ `<!DOCTYPE text [`
 - `<!ELEMENT text` 0 0 (kopf, (absatz | liste)+) `>`
 - `<!ELEMENT kopf` 0 0 (#PCDATA)
 - `<!ELEMENT absatz` - 0 (#PCDATA)
 - `<!ELEMENT liste` - - (lp)*
 - `<!ELEMENT lp` - 0 (#PCDATA)`]`
`>`
- ◆ Kopf des Textes
 - `<absatz>` 1. Absatz
 - `<liste>` `<lp>`Listenelement 1 `</liste>`
 - `<absatz>` 2. Absatz

SGML Dok.: Regeln für die Minimierung

- Ende und Start von beweglichen Elementen markieren
- Endmarkierungen minimieren bei klar definierten hierarchischen Elementen
- Im Zweifelsfall keine Minimierung
- Eine Startmarkierung kann nur dann minimiert werden, wenn das betreffende Element nicht optional ist und alle anderen an dieser Stelle möglichen Elemente optional sind
- Die leere Endmarkierung `</>` terminiert das zuletzt geöffnete Element.

SGML Dok.: Inklusionen, Exklusion

- Folgt dem Inhaltsmodell mit folgender Syntax
 - ◆ Inklusion: + (Elementname (| Elementname) *)
 - ◆ Exklusion: - (Elementname (| Elementname) *)
- Beispiele: Absatz kann Listen oder Fussnoten enthalten
 - ◆ <!ELEMENT absatz - - (#PCDATA) + (liste | fn) >
- Beispiele: Fussnote kann selbst keine Fussnote enthalten
 - ◆ <!ELEMENT fn - - ((#PCDATA) - (fn) >

SGML Inhalte.: Inhalte und Entitäten - Definitionen

- Zeichen
 - ◆ Steht für irgend etwas. Laute, Stop, Verkehrszeichen
- Buchstabe
 - ◆ Konkrete Erscheinungsform eines Zeichens. A und a sind verschiedene Buchstaben für dasselbe Zeichen.
- Schrift oder Font:
 - ◆ Times, Courier, Helvetica
- Schriftfamilie:
 - ◆ Gruppen von einander vom Aussehen her ähnlicher Schriften

SGML Inhalte.: Definitionen

- Ein Buchstabe kann auch für mehrere Zeichen stehen.
 - ◆ Bspl: Bindestrich, Minus, Gedankenstrich
- Zeichensatz:
 - ◆ eindeutige Zuordnung zwischen einer Menge von Zeichen (einem Alphabet) und einer Menge von Zahlen (Bspl: ASCII Zeichensatz)
- Zeichencode:
 - ◆ einem Zeichen zugeordnete Zahl
- Daten des Dokumentes:
 - ◆ Inhalte der im Dokument enthaltenen Elemente.

SGML Inhalte.: Ziel

- Die Darstellung von Zeichen in SGML Dokumenten soll unabhängig vom verwendeten Alphabet und unabhängig von dessen Codierung sein.
- Unabhängigkeit von Codierung
 - ◆ gewährleistet Portierbarkeit auf niedriger Ebene, mit relativ beschränktem Alphabet
 - ◆ Problem wird durch SGML-Deklaration gelöst
 - ◆ By default geht SGML vom Basiszeichensatz ISO 646 (=ASCII) aus, der für die SGML Deklaration verwendet werden muss.
 - ✦ Public Bezeichner: „ISO 646:1983//CHARSET International Reference Version (IRV)//ESC 2/5 4/0“

SGML Inhalte.: Ziel

■ Unabhängigkeit von Codierung

- ◆ Enthält auch Umsetzungstabelle, die den Codes des Dokumentzeichensatzes die entsprechenden Codes des Basiszeichensatzes zuordnet

Dok	Anz.	Basis
129	9	97
0	9	unused

SGML Inhalte.: Syntax der SGML Deklaration

■ <!SGML Version

CHARSET	Zeichensatz-Spezifikation
CAPACITY	Kapazitätsspezifikation
SCOPE	Syntax-Bereichs-Spezifikation
SYNTAX	Syntax-Spezifikation
FEATURES	Merkmal-Spezifikation
APPINFO	Anwendungsinformation
>	

SGML Inhalte.: Unabhängigkeit vom Alphabet

- Durch Einführung von „Entitäten“ kann der beschränkte Zeichenvorrat eines Alphabetes erweitert werden.
- Entität: benanntes Objekt, das ein Zeichen, eine Zeichenkette oder eine ganze Datei sein kann.
 - ◆ `<!ENTITY hyphen „-“>`
 - ◆ `<!ENTITY deg „Grad“>`
- Entitätsreferenz: *&Entitäts-Name;*
 - ◆ Bspl: `°`
- Zeichenreferenz: *&#Zeichencode*
 - ◆ kann im Text vorkommen, wird jedoch besser als Entität definiert
`<!ENTITY hyphen „-“>`

SGML Inhalte.: Unabhängigkeit vom Alphabet

- SDATA: systemabhängige Daten
 - ◆ `<!ENTITY hyphen SDATA „-“>`
- CDATA: Daten, die vom Parser ignoriert werden
 - ◆ `<!ENTITY lt CATA „<“>`
- NDATA: Keine SGML Daten
- Daten-Entitäten: CDATA, SDATA, NDATA
- Text-Entitäten: Macros oder Textbausteine für das Dokument
 - ◆ `<!ENTITY mfg „Mit freundlichen Grü&szelig;en“>`
- Externe Entitäten: externe Dateien
 - ◆ `<!ENTITY kap1 SYSTEM „c:\kap1.sgm“>`

SGML Inhalte.: Unabhängigkeit vom Alphabet

- SUBDOC-Entitäten: externe Dokumente mit eigener DTD
 - ◆ `<!ENTITY kap1 SYSTEM „c:\kap1.sgm“ SUBDOC>`
- Parameter-Entitäten: können nur in DTD verwendet werden
 - ◆ `<ENTITY %floats „em | fig | fn“>`
 - ◆ `<ENTITY em -- (#PCDATA) - (em) + (%floats;)>`
- Public entity sets: standardisierte Alphabeterweiterungen

Attribute

- Attribute qualifizieren Elemente und Elementinhalte
 - ◆ englische, deutsche oder französische Zitate
 - ◆ `<zitat sprache=„englisch“> hello, how are you? </zitat>`
- Syntax der Attribut-Deklarations-liste
 - ◆ `<!ATTLIST`
(Elementname | Elementliste)
(Attributname Wertedeklaration Voreinstellung)+ `>`
 - ◆ `<!ATTLIST bild`
format (hoch | quer) hoch
rahmen (ja | nein) ja `>`

Nichtlineare Elemente

- Verzeichnis
- Literaturverzeichnis
- Index
- Glossar
- Fussnote
- Marginalie
- Annotation
- Verweis

Notationen

- Eine Notation ist ein benanntes, auf Daten anzuwendendes Verfahren, das sowohl ein Verfahren zur Interpretation, als auch ein Verfahren zur Aufbereitung der Daten sein kann.
 - ◆ `<!NOTATION tex PUBLIC`
 `„-//LOCAL//NOTATION Tex Formula//EN“`
 `„c:\sgml\tex2bmp.exe“>`
 - ◆ `<!NOTATION tex PUBLIC`
 `„-//LOCAL//NOTATION Tex Formula//EN“ >`
 - ◆ `<!ELEMENT formel - - CDATA >`
 - ◆ `<!ATTLIST formel typ (tex | eqn) tex >`
 - ◆ `<formel>\sqrt{1 - \cos(x)} </formel>`