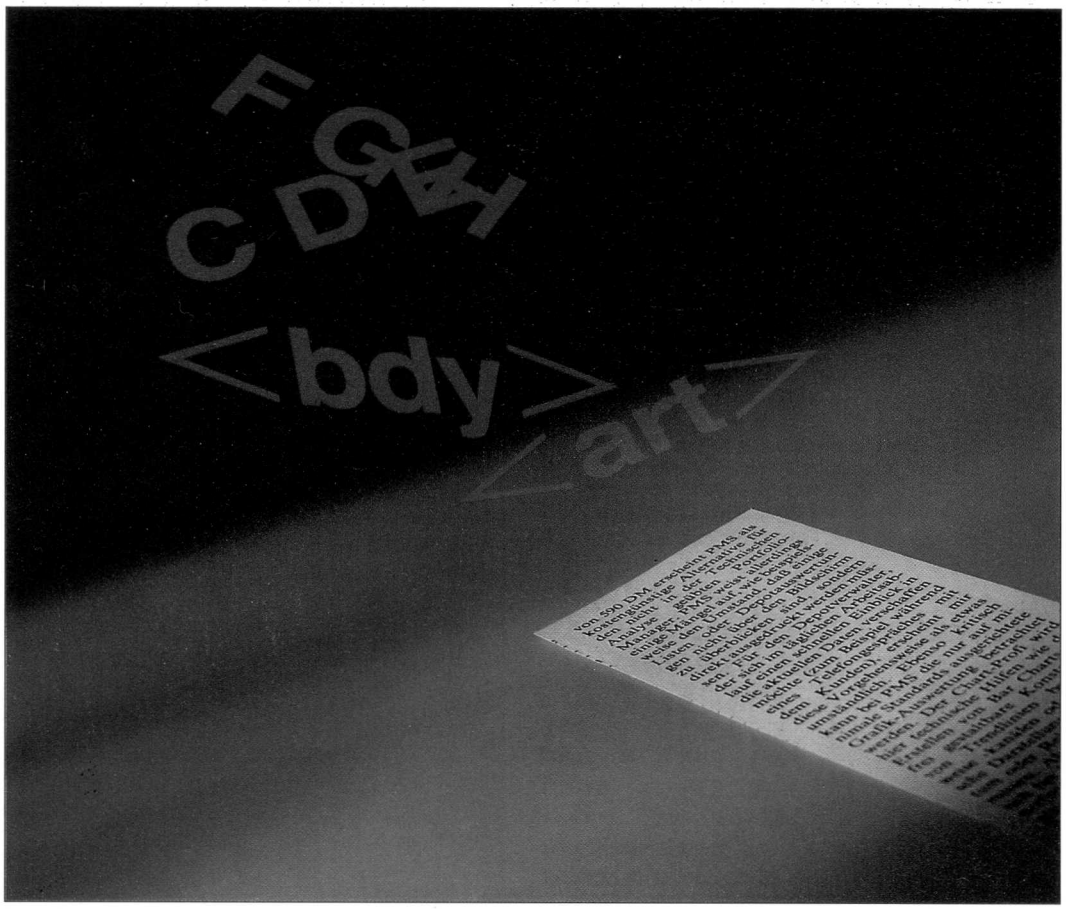


Mit spitzer Klammer

Internationaler Standard für Textdateien

Dr. Hartwig Thomas

Desktop Publisher sind fein raus; sie erzeugen mit ihrem DTP-System PostScript-Dateien, die jeder beliebige (Laser-)Drucker ausdrucken kann, der diese Seitenbeschreibungssprache versteht. Was ist aber, wenn ein Autorenprodukt nicht im Papierausdruck seine letzte Erfüllung finden soll. Im Vor-PostScript-Stadium sieht es mit seiner Austauschbarkeit düster aus; kocht doch jedes kleine oder große Textprogramm sein eigenes Süppchen: eine Prise $0E hier, eine Messerspitze $0D dort, das Ganze garniert mit einem kunterbunten Header ... Viele Geschmacksnuancen mögen den Gaumen erfreuen, dem Textproduzenten ist eher mit einem standardisierten Format gedient.

Im Dezember 1988 erklärte das amerikanische nationale Standard-Institut den Vorschlag der Association of American Publishers (AAP) eines 'Standards für elektronische Manuskript-Präparation und -Markierung' für verbindlich. Dieser hierzulande bisher unbekannte Standard für die Speicherung von Texten besteht aus zwei Hauptteilen. Zum einen enthält er eine Definition des Stoffs, aus dem die Texte sind: der Buchstaben und Zeichen. Diese werden im Standard 'Referenzen' genannt. Zum anderen legt er für einige Dokument-Typen fest, wie man ihre Struktur mit Marken in spitzen Klammern deklariert. Als Basis dienen zwei internationale Standards: die internationale Version des ASCII-Codes 'IRV' (= International Reference Version) und die Textstruktursyntax SGML (= Standard Generalized Markup Language).

Die Bemühungen der 'Electronic Publishing Special Interest Group' – EPSIG, eine Kollaboration des Online Computer Library Center (OCLC) und der amerikanischen Verleger-Vereinigung (AAP) – und anderer Verfechter des Konzepts 'Generalized Markup Language' (GML) lassen sich an einer deutlichen Konvergenz der 'Markierungs-Bezeichnungen' in verschiedenen Textsystemen beobachten: IBMs GML/DCF, Borlands Sprint, Knuths beziehungsweise Lamports LaTeX benutzen alle ähnliche Strukturmarken, die sich hauptsächlich in den syntaktischen Details unterscheiden.

Wie in Babylon

Jedem PC-Benutzer dürfte schon aufgefallen sein, daß es recht schwierig ist, einem Kollegen Texte auf Diskette mitzuteilen, wenn dessen Textverarbeitungssystem nicht mit dem eigenen identisch ist. Und es ist keine Seltenheit, daß ein Text zwischen Autor und Satzmaschine mehrfach ausgedruckt und abgetippt wird, obwohl heute ein großer Teil aller Texte in elektronischer Form auf Computern entsteht.

Den Verlagen ist nicht zuzumuten, daß sie sämtliche denkbaren Textverarbeitungsformate 'verstehen' können, zumal sich viele Textsysteme einen Sport aus der Geheimhaltung ihrer Datenformate zu machen scheinen. Obwohl die meisten Systeme einen Import/Export von/in 'ASCII-Format' als globales, standardisiertes Austauschformat anbieten, ist das absolut ungenügend für durchschnittliche Texte. Weder Textauszeichnungen ('kursiv', 'fett') noch Referenzen (z. B. Fußnoten) bleiben bei dieser Radikalvereinfachung des Textes erhalten. Nicht einmal die Absätze sind im allgemeinen aus dem ASCII-Text rekonstruierbar.

Die redaktionelle Neugestaltung dieser strukturellen Eigenschaften aus dem bloßen ASCII-Text ist kaum weniger mühevoll als das Abtippen. Ohne ausgedrucktes Typoskript ist die Rekonstruktion unmöglich. Auch die weitverbreitete Seitenbeschreibungssprache 'Postscript' taugt als Austausch-Standard nicht. Da die äußere Erscheinung des Textes in dieser Sprache schon weitgehend festgelegt ist, geht ihr die notwendige Flexibilität ab, Zusätze oder Veränderungen aufzunehmen.

Für die Kommunikation über Netzwerke, für das 'elektronische Verlagsgeschäft' etwa auf CD-ROM, für die portable Gestaltung von Nachschlageprogrammen auf großen Text-Datenbanken ist die Strukturierung einer ASCII-Datei ungenügend. Im Gegensatz zu gewissen Hardware-Entwicklungen stellt keines der heute verbreiteten Textsysteme einen

De-facto-Industriestandard dar. Nur von internationalen Standardkommissionen kann eine Vereinheitlichung des Austauschformats elektronischer Texte erwartet werden.

'ASCII and ye shall receive'

Gegen Ende der sechziger Jahre hatte sich das Byte (auch Oktett genannt), das Paket von acht Bits, als kleinste 'vernünftige' Dateneinheit bei den meisten Computer-Herstellern durchgesetzt. Es lag nahe, für die Kodierung von Texten jedem der 256 möglichen Bit-Konfigurationen eines Bytes (mit Code von 0 bis 255) einen Buchstaben oder ein Zeichen zuzuordnen.

Schon früh wurde vom amerikanischen nationalen Standard-Institut (ANSI) eine solche Zuordnung unter dem Namen ASCII (American Standard Code for Information Interchange) standardisiert. Dabei wurden nur die Bytes mit Code von 0 bis 127 mit einer Bedeutung belegt. Zum einen vermischte man damals Code mit Übertragung und reservierte das achte Bit für die Fehlererkennung (Parity Bit), zum andern konnte man sich damals wohl kaum vorstellen, wie man die vielen Zeichen sinnvoll verwenden könne, wo man doch für das ganze Alphabet in Groß- und Kleinschreibung knapp die Hälfte der ersten 128 Codes gebraucht hatte.

Eine ASCII-Datei ist also eine beliebig lange Folge von Bytes, die als Codes gemäß der ASCII-Tabelle [siehe z. B. 9] interpretiert werden. Als Steuerzeichen sind neben den Buchstaben (32 – 126) nur Tabulator (9), Zeilenvorschub (10) und Wagenrücklauf (13) in ihrer ursprünglichen Teletype-steuernden Zeilenbegrenzungs-Bedeutung und eventuell Seitenvorschub (12) für die Seiteneinteilung zugelassen. Es hat sich eingebürgert, den Wagenrücklauf vor dem Zeilenvorschub durchzuführen.

1983 erschien dann die internationale Version der elektronischen Textkodierung: ISO 646 '7-Bit Coded Character Set for Information Processing Interchange', bekannt geworden unter dem Kürzel IRV ('International Reference Version'). IRV-Codes unterschieden sich von ASCII-Codes in zwei Zeichen: Code 36 steht im ASCII für '$', im IRV für '¤', das internationale Währungszeichen, und Code 126 steht im ASCII für die Tilde ('~'), im IRV für den Überstrich ('¯'). Der auch von den Post- und Fernmeldeverwaltungs-Gremien PTT und CEPT für Videotext verwendete IRV-Standard bildet die Basis für den neuen SGML-Standard.

Vom mark-up ...

SGML, die Grundlage des neuen Manuskript-Standards, ist keine neue Erfindung, sondern der Endpunkt einer Entwicklung elektronischer Texte. Der Vorgänger von SGML hieß GML (= Generalized Markup Language), und sein Erfinder war Charles F. Goldfarb.

Als Charles F. Goldfarb gegen Ende der sechziger Jahre von IBM als Jurist angestellt wurde, war die Textproduktion dieser Abteilung des Konzerns schon sehr umfangreich. Die Akten der Anti-Trust-Prozesse füllten ganze Gebäudekomplexe. Goldfarb störte sich an der damaligen umständlichen Methode der elektronischen Textherstellung. Die Texte wurden vom Autor mit Befehlen zur Druckgestaltung gemischt. Darauf rief man ein Programm auf, das aufgrund dieser Befehle den Ausdruck produzierte. Die meisten heutigen Textverarbeitungs- und Schriftsatz-Systeme arbeiten übrigens nach genau demselben Schema. Der Unterschied ist einzig, daß man die Wirkung der Befehle schneller am Bildschirm sieht und die Eingabe der Drucksteuerungsbefehle mit Hilfe von modernen Eingabetechniken (Maus, Ikonen ...) geschieht.

Der junge Jurist wurde nach anfänglichem Ärger über die Unzulänglichkeit der Computer-Satzsysteme vom furor typographicus ergriffen. Die Probleme des Schriftsatzes scheinen seit der bahnbrechenden Erfindung der Linotype besonders in den Vereinigten Staaten den Erfindergeist zu reizen. Der Jurist Charles F. Goldfarb und der Mathematiker Donald Knuth sind als die Pioniere der Computer-Typographie in die Geschichte eingegangen.

Goldfarb hatte beobachtet, daß besonders am Anfang und am Ende des Dokuments wie am Anfang und Ende seiner Teile wilde Anhäufungen von Druckgestaltungs-Befehlen vorkamen. Er fing an, diese unter Namen wie 'Kapitelbeginn', 'Kapitelende', 'Unterabsatz', 'Beginn Zitat' und so weiter als Makros zusammenzufassen. Der Text gewann dadurch viel an Lesbarkeit und die genauen Satzoperationen für diese textlogischen Elemente konnten sauber vom eigentlichen Text getrennt bleiben.

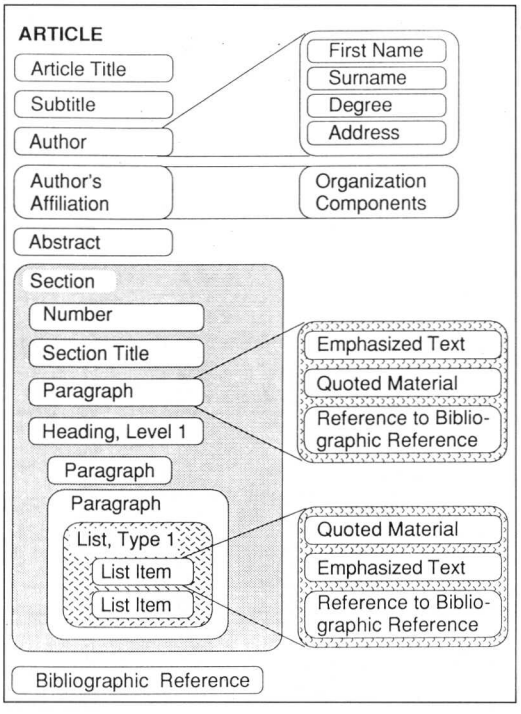
Die Makronamen näherten sich in ihrer Bedeutung den traditionellen redaktionellen Markierungen, die dem Setzer mitteilten, welche Textteile wie zu setzen waren. Dieser Vorgang hieß in der amerikanischen Verlegertradition 'mark-up'.

Elemente, die nichts mit dem Autor und der Textlogik zu tun hatten, wie etwa die Entscheidung, ob jedes Hauptkapitel auf einer ungeraden Seite anzufangen habe, waren auf diese Art an den Makro-programmierenden Setzer delegiert. Die Wiedergeburt der Arbeitsteilung auf dem Gebiet der Textproduktion wird denn auch heute noch auf der ersten Seite eines jeden IBM-Handbuchs über die Generalized Markup Language (GML) zelebriert, wo es heißt, daß es Dokument-Administratoren, Autoren, Setzer und andere gibt [1].

Goldfarb war den entgegengesetzten Weg des heutigen Desktop Publishers gegangen: Er wünschte zwar hohe Qualität seiner Publikationen, wollte aber als Autor nichts mit der – mühevollen, und zu Recht von professionellen Setzern ausgeübten – Arbeit der visuellen Gestaltung zu tun haben. Als Mitarbeiter in einem Großkonzern, wo es für jede Spezialaufgabe einen Spezialisten gibt, wäre ihm die Idee einer Satzmaschine auf der 'Schreibtisch-Oberfläche' sicher schrecklich vorgekommen.

Er wollte aber andererseits auf das bequeme Instrument des Computer-Texteditors nicht verzichten und auch alle 'strukturellen' Teile seiner Arbeit als Autor beherrschen und sie nicht dem Setzer überlassen. Die grundlegende Philosophie seiner generellen Markup-Sprache GML war also, daß sie keine Reste enthalten durfte, welche sich auf die äußere Erscheinung bezogen, sondern nur solche Angaben, die autor- und verlegerbezogen die Textlogik markierten [2].

Unter Textlogik verstand Goldfarb die Angabe, welche Rolle die verschiedenen Teile des Textes spielten (z. B. Inhaltsverzeichnis, Fußnote, Kapitelti-



Strukturdefinitionen sind hierarchisch organisiert, wie die Grafik zeigt. Jedes Element kann wieder eine Reihe untergeordneter Elemente enthalten.

tel, Bibliographie, Unterabschnitt ...) ohne Bezug, wie diese typographisch zu realisieren seien.

Der Autor hatte sich an seine Syntax der Textbeschreibung zu halten. Deren typographische Semantik war in Programmen festgelegt, welche – die GML-Marken interpretierend – den Schriftsatz durchführten. Als Substrat war für GML also grundsätzlich jedes makrofähige Textsystem brauchbar: IBMs 'SCRIPT' (bzw. heute: 'DCF') konnte ebensogut als Pfropfboden für GML dienen wie etwa Donald Knuths 'TeX', das 1978 in einer ersten Fassung vorlag und bezüglich typographischer Ausdrucksmächtigkeit noch heute alle Satzsysteme in die Tasche steckt. Die TeX-Makrosammlung 'LaTeX' von Lesley Lamport kommt einer solchen GML-Implementation sehr nahe. Brian Reids 'SCRIBE' dürfte das erste Textverarbeitungs-Programm gewesen sein, das ausschließlich auf dem GML-Konzept beruhte [3].

... zur markigen Sprache

Obwohl Goldfarbs GML-Idee innerhalb der IBM auf gewisse Resonanz stieß und sogar als IBM-Produkt realisiert wurde, war der Widerstand anderer Textprogramm-Fraktionen so groß, daß externe Abstützung von GML wichtig wurde. Es liegt wohl auch in der Natur der standardisierenden GML-Idee eines Juristen, daß sein Werk erst mit der allgemein gültigen, möglichst internationalen Standardisierung als abgeschlossen betrachtet werden kann.

Goldfarb nahm also Einsitz in der Internationalen Standard Organisation (ISO) und war beteiligt an der Formulierung des SGML-Standards ISO 8879 [4].

Da es nicht angeht, die Struktur von Texten ein für allemal verpflichtend festzulegen, wählte diese Organisation einen für das Computer-Zeitalter typischen Ausweg: sie definierte eine Syntax. Statt die allgemein zulässigen Strukturen von Texten festzulegen, standardisierte sie eine formale Metasprache, in der man Textstrukturen formulieren kann.

Der ISO-Standard führt neben dieser Strukturbeschreibungssyntax ein weiteres Konzept ein: die Öffentlichkeit gewisser Definitionen, sogenannten 'Applikationen'. In diesen Applikationen werden gewisse Dokumenttypen definiert und für 'öffentlich' erklärt. Diese Anwendungen sind für den Durchschnittsautor der eigentliche Standard, da die Metasprache des SGML doch eher für Compilerprogrammierer verwendbar ist.

Da beim Austausch von Texten dem Absender und Empfänger dieselbe Strukturbeschreibung zusammen mit dem strukturierten Text bekannt sein muß, kann man entweder auf solche 'öffentlich' bekannten Applikationen Bezug nehmen oder eine eigene in der SGML-Strukturbeschreibungssprache definierte Applikation mitschicken.

Schon seit einiger Zeit setzt sich bei den amerikanischen Verlegern eine Erkenntnis durch, die wohl nach und nach auch den deutschsprachigen Raum durchdringen wird: Verleger verkaufen Informationen, nicht Druckerschwärze auf Papier. Diese Information kann man über Leitungen, optische Platten, Videosignale und Satelliten (mehrfach) vermarkten. Damit dies möglich ist, muß sie in einer geräteunabhängigen Form vorliegen, welche die optische Gestaltung möglichst wenig präjudiziert.

Es sind daher auch die Verleger, die nach der Publikation des ISO-Standards von SGML [5] die Verleger-Applikation 'Standard for Electronic Manuscript Preparation and Markup' [6] vorantrieben, der im Dezember 1988 zum ANSI/NISO-Standard Z39.59-1988, dem 'American National Standard for Electronic Manuscript Preparation and Markup' avancierte. Die Electronic Publishing Special Interest Group (EPSIG) vertreibt seither neben einer Reihe von einführenden Publikationen zu dem neuen Standard die von AAP veröffentlichten Standard-Dokumente 'Buch' (Book: BK-1), 'Artikel' (Article: ART-1) und 'Reihe' (Serial: SER-1) [7].

STRUCTURAL DEFINITION FOR ARTICLE

Article (<article>)*
(Elements from List 1 are allowed anywhere in article unless otherwise stated)
Front matter (<fm>) (No start-tag or end-tag required)
Title group (<tig>) (No start-tag or end-tag required)
Article title (<atl>)
Subtitle (<sbt>) (optional)
Author (<au>) (optional and repeatable)
Corporate body as an author (<cau>) (optional and repeatable)
Publisher's front matter group (<pubfm>) (optional)
Any of the following, in any order, and repeated if necessary:
- Contract/grant sponsor (<cgs>)
- Contract/grant number (<cgn>)
- Copyright notice (<crt>)
- Reprint source (<rps>)
- Distributor/available from (<avl>)
- Acid free paper indicator (<phi>)
- Price (<prc>)
- Extent of work (<ext>)
- CODEN (<cdn>)
- Acquisition/order number (<aon>)
- Article identifier (<aid>)
- ISSN (<issn>)

(Other front matter elements; all are optional and may be entered in any order)
Dedication (<ded>)
Abstract (<abs>)
Supporting material - availability (<smtl>)

Body matter (<bdy>) (No start-tag or end-tag required)
Section (<sec>) (one or more occurrences)
Number (<no>) (optional)
Section title (<st>) (No start-tag required if this is the first element in section)
Optionally, any elements, including repeated elements, from List 10
Subsection, level 1 (<ss1>) (zero or more occurrences)
Number (<no>) (optional)
Section title (<st>) (No start-tag required if this is the first element in subsection)
Optionally, any elements, including repeated elements, from List 10
Subsection, level 2 (<ss2>) (zero or more occurrences)
Number (<no>) (optional)
Section title (<st>) (No start-tag required if this is the first element in subsection)
Optionally, any elements, including repeated elements, from List 10
Subsection, level 3 (<ss3>) (zero or more occurrences)
Number (<no>) (optional)
Section title (<st>) (No start-tag required if this is the first element in subsection)
Optionally, any elements, including repeated elements, from List 10

Appendix matter (<appm>) (optional)
Appendix (<app>) (one or more occurrences)
Number (<no>) (optional)
Appendix title (<apt>) (No start-tag required if this is the first element in appendix)
Optionally, any elements, including repeated elements, from List 10
Section (<sec>) (zero or more occurrences)
See Section under Body matter above for definition

Back matter (<bm>) (optional)
(Any of the following, entered in any order)
Acknowledgments (<ack>)
Bibliography (<bib>)
Vita (<vt>)

*End-tag is required.

Die 'öffentliche' Strukturdefinition 'Artikel' (Article: ART-1), wie von der AAP definiert

Die AAP-Applikation

Worin besteht nun der AAP-Manuskript-Standard? Textdateien gemäß diesem Standard sind ASCII-Dateien. Genauer gesagt: IRV-Dateien. Die Grundbausteine sind also die Zeichen und Codes aus dem oben erwähnten ISO-646-Zeichensatz, welcher dem ASCII-Zeichensatz sehr nahe verwandt ist.

Damit sich die Enttäuschung umlautgeplagter Computerbenutzer in Grenzen hält: es gibt eine SGML-Technik, mit welcher man jeden europäischen Buchstaben schreiben kann, nämlich Referenz-Einheiten. Diese beginnen mit einem '&' und werden mit einem ';' abgeschlossen. Wenn ein Leerzeichen folgt, kann man das Semikolon auch weglassen.

Das Wesen des SGML-Standards wäre verfehlt, wenn sich nun der deutsche PC-Benutzer für verpflichtet hielte, 'Zürich' statt 'Zürich' zu schreiben. Nur für den standardisierten Austausch von Dateien ist der Zeichensatz auf die offiziellen 7 Bit beschränkt. Es ist aber wohl Sache der SGML-Textverarbeitung, 'Ä' in der Textdatei zu speichern, am Schirm jedoch 'Ä' anzuzeigen und 'Ä' auf dem Drucker auszudrucken. Sollte dem Autor jedoch das 'Ä' auf der Tastatur

fehlen, wird er gerne zu 'Ä' Zuflucht nehmen.

Für griechische und kyrillische Buchstaben hat man denn auch neben den Referenz-Einheiten (α β γ ...) Transliterationen definiert. Zwischen den Beginn- und End-Marken für das entsprechende Alphabet werden die eingegebenen Zeichen über die definierten Tabellen umgesetzt. Um zusammenhängende Texte in diesen Sprachen zu schreiben, ist diese Methode sehr viel praktischer als die Referenz-Einheiten.

Neben den Referenzen braucht der Autor nur noch die Textstruktur-Marken zu kennen. Diese sind alle zwischen spitzen Klammern anzugeben. Jeder Textstruktur-Bestandteil hat einen Namen, der in der SGML-Dokument-Typ-Deklaration festgelegt ist. So steht etwa 'ti' für Titel. Der Beginn der Struktur ist mit dem Strukturnamen in spitzen Klammern anzugeben: <ti> für 'hier beginnt der Titel'. Das Ende der Struktur ist durch die gleiche Marke, ergänzt um einen Schrägstrich vor dem Strukturnamen, gekennzeichnet: </ti>. Sofern der Beginn der nächsten Struktur das Ende der vorherigen impliziert, braucht dieses nicht ausdrücklich angegeben zu werden (Kapitelbeginn impliziert etwa Absatzende davor).

Um überflüssige Markiererei auf ein Minimum zu reduzieren, sind Leerzeilen statt <p> ('Paragraph' = Absatz), Anführungszeichen statt <q> und </q> ('quoted material' = Zitat), zwei Bindestriche statt '—' (Gedankenstrich) und ähnliche Abkürzungen als 'Kurzreferenzen' definiert.

Die drei großen E's

Wie man uns in der Volksschule beigebracht hat, besteht jeder Text aus den drei großen E (Einleitung, Erzählung, Ende). Gemäß dem AAP-Standard für Autoren ist ein Manuskript baumförmig zu organisieren: ein Artikel besteht aus einem Vorderteil ('front matter'), dem eigentlichen Text ('body') und eventuell gewissen Anhängseln ('back matter, appendix'). Der Vorderteil besteht wiederum aus Artikeltitel, möglicherweise mehreren Untertiteln und Autoren (oder keinem) und eventuell einem Abstract. Die Autorenbezeichnung kann dann weiter nach Namen, Vornamen, Titel, institutionelle Zugehörigkeit und Adresse aufgeschlüsselt sein.

Der eigentliche Text besteht aus einer Folge von einem oder mehreren Abschnitten ('section'). Ein Abschnitt besteht aus (optionaler) Abschnittsnummer, Abschnittstitel, einer Reihe von Absätzen, eventuell einem Untertitel der ersten Stufe, gefolgt von einer Reihe von Absätzen ... Überall im Text kann man Kommentare einfügen, die nicht zum Text zählen. Diese sind mit '<!–' und '–>' einzurahmen.

Obwohl sich das alles ein bißchen kompliziert anhört, kann ein sauber markierter Artikel doch recht lesbar sein. Grafische Übersichten über die Struktur der drei definierten Texttypen vereinfachen die Arbeit des Autors zusätzlich. Das praxisorientierte Autorenhandbuch von EPSIG [7, Author's Guide] hat denn auch nur fünfzig Seiten, wovon mehr als die Hälfte für die Liste der Spezialzeichen, den IRV-Zeichensatz und Beispiele verwendet ist.

Character	*Entity Reference*	*Description*
á	á	small a with acute accent
Á	Á	capital A with acute accent
à	à	small a with grave accent
À	À	capital A with grave accent
â	â	small a with circumflex accent
Â	Â	capital A with circumflex accent
ä	ä	small a with diaeresis or umlaut mark
Ä	Ä	capital A with diaeresis or umlaut mark
ã	ã	small a with tilde
Ã	Ã	capital A with tilde
ă	ă	small a with breve
Ă	Ă	capital A with breve
å	å	small a with ring
Å	Å	capital A with ring
ā	ā	small a with macron
Ā	Ā	capital A with macron
ą	ą	small a with ogonek
Ą	Ą	capital A with ogonek
æ	æ	small ae diphthong
Æ	Æ	capital AE diphthong
ć	ć	small c with acute accent
Ć	Ć	capital C with acute accent
ĉ	ĉ	small c with circumflex accent
Ĉ	Ĉ	capital C with circumflex accent

Mit den Referenz-Einheiten erreicht man auch alle Zeichen außerhalb der 7-Bit-ASCII-Tabelle.

SGML jetzt

Solange die heute verbreiteten größeren Textverarbeitungs-Systeme SGML noch nicht offiziell unterstützen, kann man die Austauschbarkeit von Texten maximieren, wenn man sich an die Kurzreferenzen und in etwa an eine Strukturdefinition hält. Einheitliche Markierung gleichartiger Textteile ist sogar dann schon nützlich, wenn es sich nicht um die offiziellen SGML-Markierungen handelt.

In makrofähigen Systemen (Sprint, TeX ...) sollte man Bereiche und Bezeichnungen soweit SGML-gerecht definieren, wie das vom System aus möglich ist. Generell sind keine Steuerbefehle für die äußere Erscheinung zu verwenden, sondern Textstruktur-Befehle.

Auf längere Sicht kann einem Autor nur zugemutet werden, SGML-Text zu produzieren, wenn er mindestens ein System hat, das seine Syntax prüft. Noch besser wird ein Textsystem sein, das die SGML-konforme Speicherung selbst übernimmt. Verwendung von SGML bedeutet keinen Verzicht auf WYSIWYG: Auch wenn die Datenstrukturen konvergieren, wird die Vielfalt der Schnittstellen bleiben.

Da im Zusammenhang mit Textdatenbanken, CD-ROM-Publikationen und in Kommunikation mit großen amerikanischen Verlegern der neue Standard bald unumgänglich wird, dürfte sich der Druck auf die Hersteller von Textsystemen verstärken, so daß die meisten Systeme neben dem ASCII-Export über kurz oder lang auch eine SGML-Schnittstelle erhalten. Diese Umstellung wird aufgrund der Logik der Benutzerschnittstelle je nach Textsystem mehr oder weniger Mühe bereiten. Schon heute gibt es eine Reihe von SGML-verarbeitenden Programmen [8].

... und zukünftig

Im März 1989, also wenige Monate nach der Sanktionierung des AAP-Standards durch das amerikanische nationale Standard-Institut (ANSI) propagierten Vertreter von EPSIG den neuen Standard an der CD-ROM-Konferenz in Los Angeles an einem eigenen Stand. Noch während der Ausstellung war eine sichtbare Ausbreitung von SGML von Stand zu Stand zu beobachten: verschiedenste CD-ROM-Hersteller gaben vor, 'auch', 'bald' oder 'schon' SGML zu unterstützen.

Auf der CD-ROM-Expo in Washington (3./4. Okt. 1989) war SGML schon eines der acht 'in-depth tutorials' für CD-ROM-Verleger und kam noch in einem weiteren Konferenzbeitrag zum Zuge.

Für die kluge Speicherung, Recherche und Weiterverarbeitung von Texten ist der SGML-Standard ein Segen. Dank SGML wird die Integration von Texten in eine große Datenbank kaum mehr Aufwand erfordern. Dank der Strukturierung der Texte, kann die Abfrage gezielt erfolgen. Die dem Text gemäße Abfrage-Struktur ist ein Abbild seiner Textstruktur. Ein einziges SGML-Fenster auf verschiedenste CD-ROM-Publikationen verwandelt das CD-ROM-Laufwerk vom Buch in eine Bibliothek. In den meisten Fällen sind textbasierte flexible Retrievalsysteme schon heute angemessener als die verbreiteten dBase-Dinosaurier.

Momentan werden von verschiedensten Interessengruppen neue Dokumenttypen in SGML

```
<article>   <!-- Dies ist ein Beispiel für SGML-Markup gemaess ART-1 -->
<atl>Lauter spitze Klammern
<sbt>Ein neuer Text-Standard f&uuml;r Autoren und Verleger
<au><fnm>Hartwig<snm>Thomas<deg>Dr. sc. math.
<aff><onm>Enter AG<str>Fabrikstrasse 30<pc>CH-8005<cty>Z&uuml;rich<cny>Schweiz

<abs>
Im Dezember des letzten Jahres hat das amerikanische nationale Standard-Institut (ANSI)
den SGML-Manuskript-Standard der Amerikanischen Verleger-Vereinigung AAP zum nationalen
Standard erhoben. Aus diesem Anlass wird im Folgenden dargelegt, worum es bei diesem
Standard f&uuml;r "elektronische" Manuskripte geht. Seine Entwicklung, der heutige Stand
und die Zukunftsaussichten von SGML (Standard Generalized Markup Language) werden
beleuchtet.
</abs>

<bdy>
Im Dezember 1988 erkl&auml;rte das amerikanische nationale Standard-Institut den Vorschlag
der <it>Association of American Publishers (AAP)</it> eines "Standards f&uuml;r
elektronische Manuskript-Pr&auml;paration und -Markierung" f&uuml;r verbindlich. Dieser
hierzulande nahezu v&ouml;llig unbekannte Standard f&uuml;r die Speicherung von Texten
wird wohl in naher Zukunft eine grosse Verbreitung finden. Der neue Standard besteht aus
zwei Hauptteilen. Zum einen enth&auml;lt er eine Definition des Stoffs, aus dem die Texte
sind: der Buchstaben und Zeichen. Diese werden im Standard "Referenzen" genannt. Zum
anderen legt er f&uuml;r einige Dokument-Typen fest, wie man ihre Struktur markiert. Als
Basis dienen zwei internationale Standards: die internationale Version des ASCII-Codes
"IRV" (= International Reference Version) und die Textstruktursyntax SGML (= Standard
Generalized Markup Language).

Der neue Manuskript-Standard ist zwar noch nicht sehr bekannt, die Bem&uuml;hungen der
<it>"Electronic Publishing Special Interest Group" (EPSIG)</it><nit>Eine Kollaboration des
<it>Online Computer Library Center (OCLC)</it> und der <it>Association of American
Publishers (AAP)</it>.</nit> der amerikanischen Verleger-Vereinigung und anderer
Verfechter des Konzepts "Generalized Markup Language" (GML) liessen sich jedoch an einer
deutlichen Konvergenz der "Markierungs-Bezeichnungen" in verschiedenen Textsystemen
beobachten: IBM's GML/DCF, Borlands Sprint, Knuths bzw. Lamports LaTEX ben&uuml;tzen
alle &auml;hnliche Strukturmarken, die sich haupts&auml;chlich in den syntaktischen
Details unterscheiden.

<sec>
Wozu &uuml;berhaupt Standardisierung von elektronischen Texten?

Jedem PC-Ben&uuml;tzer d&uuml;rfte schon aufgefallen sein, dass es recht schwierig ist,
Texte einem Kollegen auf Diskette mitzuteilen,

. . .

Nachschlage-Handbuch der Satz: "Autoren, Verleger, Typographen und sekund&auml;re
Informationsdienste k&ouml;nnen anfangen, den Elektronischen Manuskript-Standard zu
ben&uuml;tzen im Bewusstsein, dass seine Form, wie diejenige eines jeden Standards, nie
v&ouml;llig endg&uuml;ltig sein wird." Mit der Standardisierungs-Kommission hoffen wir,
dass SGML, wie alle guten Standards, im Gebrauch t&auml;glich besser wird.
</article>
```

So ist ein Artikel im SGML-Standard gespeichert, das heißt jedoch nicht, daß er sich auch so am Bildschirm präsentieren muß – hier sind die Textsysteme gefordert.

definiert (Dissertation, Technisches Handbuch ...). Für Autoren und Verleger deutscher Sprache wäre es vielleicht an der Zeit, ihre eigenen Dokumentdefinitionen 'öffentlich' zu machen. Die zu erwartende Reduktion an Umsetzungsarbeit alleine müßte diesen Schritt schon attraktiv erscheinen lassen.

Zur Frage der Endgültigkeit von solchen Standards findet sich im Vorwort zum Nachschlage-Handbuch der Satz: 'Autoren, Verleger, Typographen und sekundäre Informationsdienste können anfangen, den Elektronischen Manuskript-Standard zu benutzen, im Bewußtsein, daß seine Form, wie diejenige eines jeden Standards, nie völlig endgültig sein wird.' Mit der Standardisierungs-Kommission hoffen wir, daß SGML, 'wie alle guten Standards, im Gebrauch täglich besser wird'. (bb)

Literatur

[1] Charles F. Goldfarb, Document Composition Facility Generalized Markup Language: Concepts and Design Guide, Form No. SH20-9188-0, IBM Corporation, White Plains, 1980

[2] Charles F. Goldfarb, E. J. Mosher, T. I. Peterson, An Online System for Integrated Text Processing, Proceedings of the American Society for Information Science, 7, 147 – 150 (1970)

Charles F. Goldfarb, A Generalized Approach to Document Markup, in: Textverarbeitung und Bürosysteme, IBM Informatik Symposium, Hg. A. Endres, J. Reetz, R. Oldenbourg Verlag, München, 1982

[3] Brian K. Reid: The Scribe Document Specification Language and its Compiler, Proceedings of the International Conference on Research and Trends in Document Preparation Systems, 59 – 62 (1981)

[4] Weitere Literatur zu SGML:

Joan M. Smith und Robert Stutely, SGML: The User's Guide to ISO 8879, Ellis Horwood, Chichester, England, 1988, Vertrieb: Halsted Press

Martin Bryan, SGML: An Author's Guide to the Standard Generalized Markup Language, Addison-Wesley, Wokingham, England, 1988. Techniques for Using SGML, ISO/IEC/TR9573

[5] ISO 8879, Information processing – Text and Office Systems – Standard Generalized Markup Language (SGML), 1986

[6] Electronic Manuscript Project: Standard for Electronic Manuscript Preparation and Markup, Association of American Publishers, Electronic Manuscript Series, 1987

Electronic Manuscript Project: Markup of Tabular Material, Association of American Publishers, Electronic Manuscript Series, 1987

Unter den Sponsoren des Projekts:

CBS Publishing, Elsevier Science Publishing Co., Harper & Row Publishers Inc., Hearst Books, IEEE, Macmillan Publishing Co., McGraw-Hill Book Company, New England Journal of Medicine, Prentice-Hall Inc., John Wiley & Sons Inc, Xerox Corporation. Addison Wesley Publishing Co., Pergamon Press Inc, Rand McNally & Company, Simon & Schuster Inc, Springer-Verlag, Times Mirror ... American Mathematical Society, Cambridge University Press ...

[7] EPSIG, Author's Guide to Electronic Manuscript Preparation and Markup, Association of American Publishers, Electronic Manuscript Series, 1989

EPSIG, Reference Manual on Electronic Manuscript Preparation and Markup, Association of American Publishers, Electronic Manuscript Series, 1989

EPSIG, Markup of Mathematical Formulas, Association of American Publishers, Electronic Manuscript Series, 1989

EPSIG, Markup of Tabular Material, Association of American Publishers, Electronic Manuscript Series, 1989

AAP, Document Type Definitions and Short References, Version 2.0, die Definitionen des Manuskript-Standard auf 5.25'-Floppy-Disk, Association of American Publishers, August 1987

Alle diese Dokumente sind erhältlich bei:

EPSIG (MC 278), c/o OCLC, 6565 Frantz Road, Dublin, OH 43017-0702, USA

[8] Unter anderen:

ArborText Inc.,The Publisher 2.0, auf Sun-Workstations, Ann Arbor, Michigan, USA

IBM, SGML Translator DCF Edition, auf IBM-370-Systemen

Sobemap, MARK-IT, in C auf UNIX oder MSDOS (eingebaut in Texet, Interleaf, Scribe, Xyvision), Brüssel, Belgien, 1988

[9] Zeichensatz-Tabellen, Kartei, c't 2/88, S. 203