Ein neuer Text-Standard für Autoren und Verleger:

Lauter spitze Klammern

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 für „elektronische" Manuskripte geht. Seine Entwicklung, der heutige Stand und die Zukunftsaussichten von SGML (Standard Generalized Markup Language) werden beleuchtet.

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 nahezu völlig unbekannte Standard für die Speicherung von Texten wird wohl in naher Zukunft eine grosse Verbreitung finden. Der neue Standard 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).

Der neue Manuskript-Standard ist zwar noch nicht sehr bekannt, die Bemühungen der *Electronic Publishing Special Interest Group (EPSIG)*[1] 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 benützen alle ähnliche Strukturmarken, die sich hauptsächlich in den syntaktischen Details unterscheiden.

1 Eine Kollaboration des *Online Computer Library Center (OCLC)* und der *Association of American Publishers (AAP)*.

```
---------------------------------------------------------------------------------------
<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.

<fig>
<fgb>
&lt;article> &lt;atl>Lauter spitze Klammern
&lt;sbt>Ein neuer Text-Standard f&amp;uuml;r Autoren und Verleger
&lt;au>&lt;fnm>Hartwig&lt;snm>Thomas&lt;deg>Dr. sc. math.
&lt;aff>&lt;onm>Enter AG&lt;str>Fabrikstrasse
...
</fgb>
<fgc>So ist ein Artikel im neuen SGML-Standard gespeichert
</fig>

<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>
---------------------------------------------------------------------------------------
```

Fig. 1: So ist ein Artikel im neuen SGML-Standard gespeichert

Wozu überhaupt Standardisierung von elektronischen Texten?

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

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 ins *ASCII-Format* als globales, standardisiertes Austauschformat anbieten, ist das absolut ungenügend für durchschnittliche Texte. Weder Textauszeichnungen („kursiv", „fett") noch Referenzen („siehe auch Seite 85") 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. Die Software-Industrie wird die sich daraus für sie ergebenden Vereinfachungen dankbar aufgreifen.

„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 Codierung von Texten jedem der 256 möglichen Bitkonfigurationen 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.

Die Grosscomputer von IBM sind wohl die einzigen Maschinen, die sich bis heute standhaft diesem Code widersetzen: die Textcodierung auf diesen Maschinen geschieht nach dem IBM-Standard EBCDIC (Extended Binary Coded Decimal Interchange Code), der alle 256 Zeichen belegt und leider keine eindeutige Übersetzung in oder von ASCII erlaubt.

Wenn wir uns die ASCII-Tabelle näher betrachten, stellen wir fest, dass eine weitere Vermischung in diesen Code eingegangen ist: die eigentlichen Zeichen belegen die Plätze 32 bis 126, während die ersten 32 Codes (0-31) und der letzte (127) mit unverständlichen mehrbuchstabigen Abkürzungen bezeichnet ist. Von den ursprünglich intendierten Bedeutungen dieser „Steuerzeichen" (SOH = Start Of Header, STX = Start TeXt, ETX = End TeXt ...) sind für das, was wir im Folgenden als ASCII-Dateien bezeichnen, nur HT (ASCII 9, Horizontal Tab = Tabulator), LF (ASCII 10, Line Feed = Zeilenvorschub), FF (ASCII 12, Form Feed = Seitenvorschub) und CR (ASCII 13, Carriage Return = Wagenrücklauf) von Interesse. Sie haben keine eigentliche Zeichenbedeutung, sondern steuern den Teletype, der damals als Terminal diente.

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

Zeilenvorschub durchzuführen. Manchmal wird daher der Code 13 als Zeilenend-Code und Code 10 als Zeilenbeginn-Code bezeichnet.

Deutschsprachigen Betrachtern der ASCII-Tabelle wird kaum entgangen sein, dass man bei der Definition der ASCII-Tabelle nicht an deutschsprachige Texte gedacht hat: Weder Gänsefüsschen noch Umlaute oder Eszett sind vorgesehen. Aber wozu haben wir denn die noch unbelegten Codes l28-255? Verschiedenste Computer-Hersteller belegten diesen oberen Code-Bereich mit Umlauten und ähnlichen Spezialzeichen. Andere wiederum beschlossen, dass die eckigen und geschweiften Klammern für deutschsprachige Computerbenützer nicht so wichtig sind. Ausserdem sind ihre Tasten genau dort, wo wir unsere Umlaute haben: Ein „ü“ für eine „[“ oder umgekehrt ist auch auf heutigen PC-Druckern keine Seltenheit. Die frühen PCs „lösten“ das Problem, indem sie die Buchstaben-Bilder frei konfigurierbar machten. Da jeder Sirius-Besitzer seine eigenen Buchstaben malte, drohte das Buchstabencode-Chaos ausserhalb der recht stabilen ASCII-Codes 32-126 total zu werden.

Code	ASCII	IRV	Code	ASCII	IRV	Code	ASCII	IRV	Code	ASCII	IRV
0	NUL		16	DLE		32		[5]	48	0	0
1	SOH		17	DC1		33	!	!	49	1	1
2	STX		18	DC2		34	"	"	50	2	2
3	ETX		19	DC3		35	#	#	51	3	3
4	EOT		20	DC4		36	$	¤[6]	52	4	4
5	ENQ		21	NAK		37	%	%	53	5	5
6	ACK		22	SYN		38	&	&	54	6	6
7	BEL		23	ETB		39	'	'	55	7	7
8	BS		24	CAN		40	(	(	56	8	8
9	HT	TAB[2]	25	EM		41	)	)	57	9	9
10	LF	LF[3]	26	SUB		42	*	*	58	:	:
11	VT		27	ESC		43	+	+	59	;	;
12	FF		28	FS		44	,	,	60	<	<
13	CR	CR[4]	29	GS		45	-	-	61	=	=
14	SO		30	RS		46	.	.	62	>	>
15	SI		31	US		47	/	/	63	?	?

2 Tabulator

3 Zeilenvorschub

4 Wagenrücklauf

5 Leerschlag

6 internationales Währungssymbol

Code	ASCII	IRV	Code	ASCII	IRV	Code	PC-ASCII	Code	PC-ASCII
64	@	@	96	`	`	128	Ç	160	á
65	A	A	97	a	a	129	ü	161	í
66	B	B	98	b	b	130	é	162	ó
67	C	C	99	c	c	131	â	163	ú
68	D	D	100	d	d	132	ä	164	ñ
69	E	E	101	e	e	133	à	165	Ñ
70	F	F	102	f	f	134	å	166	ª
71	G	G	103	g	g	135	ç	167	º
72	H	H	104	h	h	136	ê	168	¿
73	I	I	105	i	i	137	ë	169	⌐
74	J	J	106	j	j	138	è	170	¬
75	K	K	107	k	k	139	ï	171	½
76	L	L	108	l	l	140	î	172	¼
77	M	M	109	m	m	141	ì	173	¡
78	N	N	110	n	n	142	Ä	174	«
79	O	O	111	o	o	143	Å	175	»

Code	ASCII	IRV	Code	ASCII	IRV	Code	PC-ASCII	Code	PC-ASCII
80	P	P	112	p	p	144	É	176	░
81	Q	Q	113	q	q	145	æ	177	▒
82	R	R	114	r	r	146	Æ	178	▓
83	S	S	115	s	s	147	ô	179	│
84	T	T	116	t	t	148	ö	180	┤
85	U	U	117	u	u	149	ò	181	╡
86	V	V	118	v	v	150	û	182	╢
87	W	W	119	w	w	151	ù	183	╖
88	X	X	120	x	x	152	ÿ	184	╕
89	Y	Y	121	y	y	153	Ö	185	╣
90	Z	Z	122	z	z	154	Ü	186	║
91	[	[	123	{	{	155	¢	187	╗
92	\	\	124	|	|	156	£	188	╝
93	]	]	125	}	}	157	¥	189	╜
94	^	^	126	~	‾[7]	158	₧	190	╛
95	_	_	127	DEL	DEL	159	ƒ	191	┐

7 Überstrich

Code	PC-ASCII	Code	PC-ASCII	Code	PC-ASCII	Code	PC-ASCII
192	└	208	╨	224	α	240	≡
193	┴	209	╤	225	ß	241	±
194	┬	210	╥	226	Γ	242	≥
195	├	211	╙	227	π	243	≤
196	─	212	╘	228	Σ	244	⌠
197	┼	213	╒	229	σ	245	⌡
198	╞	214	╓	230	µ	246	÷
199	╟	215	╫	231	τ	247	≈
200	╚	216	╪	232	Φ	248	°
201	╔	217	┘	233	Θ	249	∙
202	╩	218	┌	234	Ω	250	·
203	╦	219	█	235	δ	251	√
204	╠	220	▄	236	∞	252	ⁿ
205	═	221	▌	237	φ	253	²
206	╬	222	▐	238	ε	254	■
207	╧	223	▀	239	∩	255	

Fig. 2: ASCII-Codes, IRV-Codes, „PC-ASCII“-Erweiterung

Zur allgemeinen Überraschung unterstützte der IBM-PC, obwohl er von IBM stammte, den ASCII-Code. Zur allgemeinen Enttäuschung offerierte er keine freie Konfiguration der Buchstaben-Bilder. Der erste Ärger über den total graphiklosen PC verflog aber, als sich langsam im Laufe der Jahre dank diesem de facto Industrie-Standard des „erweiterten PC-ASCII“ fast alle Drucker dazu herab liessen, Umlaute zu drucken, wenn man diese am Bildschirm eingetippt hatte. Die unschätzbare Gewissheit des Programmierers, dass jemand „Ä“ getippt hat, wenn das Programm den Code für „Ä“ empfängt, wird erst wieder im DOS 3.3 unterminiert: Die Zeichentabellen können theoretisch vorn Benützer frei umdefiniert werden. Hoffentlich wird so etwas nicht auch noch von Software unterstützt!

1983 erschien dann die internationale Version dieses harten Kerns der elektronischen Textcodierung: 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 der PTT und CEPT für Videotext verwendete IRV-Standard bildet die Basis für den neuen SGML-Standard.

Der Erfinder von GML: Charles F. Goldfarb

SGML, der neue Manuskript-Standard 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 damals vorn 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 dass die Eingabe der Drucksteuerungsbefehle mit Hilfe von modernen Eingabetechniken (Maus, Ikonen ...) erfolgt.

Der junge Jurist wurde nach anfänglichem Ärger über die Unzulänglichkeit der Computer-Satzsysteme vorn 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, 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 an Anfang und Ende seiner Teile wilde Anhäufungen von Druckgestaltungs-Befehlen vorkamen. Er fing an, diese unter Namen wie „Kapitelbeginn“, „Kapitelende“, „Unterabsatz“, „Beginn Zitat“ u.s.w. als Makros zusammenzufassen. Der Text gewann dadurch viel an Lesbarkeit und die genauen Satzoperationen für diese textlogischen Elemente konnten sauber vorn 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 traditionellen 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 erklärt wird, dass es Dokument-Administratoren, Autoren, Setzer u.s.w. gibt[8].

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 einern Grosskonzern, wo es für jede Spezialaufgabe einen Spezialisten gibt, wäre ihm die Idee einer Satzmaschine auf der Schreibtisch-Oberfläche (desk top!) sicher schrecklich vorgekommen. Er wollte aber andrerseits 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 (Generalized Markup Language = 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[9].

Unter Textlogik verstand Goldfarb die Angabe, welche Rolle die verschiedenen Teile des Textes spielten (z.B.: Inhaltsverzeichnis, Fussnote, Kapiteltitel, 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 ebenso gut 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[10].

Der neue SGML-Standard

Obwohl Goldfarbs GML-Idee innerhalb der IBM auf gewisse Resonanz stiess und sogar als IBM-Produkt realisiert wurde, war der Widerstand anderer Textprogramm-Faktionen so gross, dass externe Abstützung

---

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

9 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. Oldenburg Verlag, München, 1982.

10 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).

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[11].

Da es nicht angeht, die Struktur von Texten im Allgemeinen 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, wurde eine formale Metasprache standardisiert, in der man Textstrukturen formulieren kann.

Der ISO-Standard führt neben dieser Strukturbeschreibungssyntax ein weiteres Konzept ein: die *Öffentlichkeit* gewisser Definitionen. Da ja beim Austausch von Texten dem Absender und Empfänger dieselbe Strukturbeschreibung zusammen mit dem strukturierten Text bekannt sein muss, kann man sie entweder in der SGML-Strukturbeschreibungssprache mitschicken oder auf „öffentlich" bekannte Strukturbeschreibungen bezugnehmen.

Im Terminalraum des Rechenzentrums der Universität Zürich steht ein amerikanisches GML-Handbuch, dessen Seitennumerierung und Textteilnamen deutsch sind („Seite 57", „Teil 1: The Front Matter", „Inhalt" ...), obwohl der gesamte Text sonst englisch ist und aus den USA stammt. Offensichtlich war die Strukturbeschreibung, mit welcher das Handbuch ausgedruckt wurde, schon auf deutsche Texte angepasst und zum Zeitpunkt des Druckens nicht mehr identisch mit der intendierten Strukturbeschreibung des Autors.

Um solche Pannen auf ein Minimum zu reduzieren, kann man basierend auf dem SGML-Standard sogenannte „Applikationen" definieren. 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.

Die *Association of American Publishers (AAP)* hat einen solchen Manuskript-Standard als SGML-Anwendung für die Texttypen „Buch" (Book: BK-l), „Artikel" (Article: ART-l) und „Reihe" (Serial: SER-l) vorgelegt. Schon seit einiger Zeit setzt sich bei den amerikanischen Verlegern eine Erkenntnis durch, die wohl im nächsten Jahrtausend auch im deutschsprachigen Raum zum Tragen kommen 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[12] die Verlegerapplikation „Standard for Electronic Manuscript Preparation and Markup"[13] vorantrieben, der im Dezem-

---

11 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.

12 *ISO 8879, Information Processing – Text and Office Systems – Standard Generalized Markup Language (SGML)*, 1986

13 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,

ber 1988 zum ANSI/NISO-Standard Z39.59-l988, dem „American National Standard for Electronic Manuscript Preparation and Markup“ avancierte. Die *Electronic Publishing Special Interest Group (EPSIG)* vertreibt seither die Standard-Dokumente (auf Wunsch auch auf Diskette) und eine Reihe von einführenden Publikationen zu dem neuen Standard[14].

Die AAP-Applikation des SGML-Standards für Verleger und Autoren

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 *Einheiten-Referenzen*. Diese beginnen mit einem „&“ und werden mit einem „;“ abgeschlossen. Wenn ein Leerschlag folgt, kann man den Strichpunkt auch weglassen.

Simon & Schuster Inc., Springer-Verlag, Times Mirror ... American Mathematical Society, Cambridge University Press, ...

14 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

AAP, *Document Type Definitions and Short References, Version 2.0*, die Definitionen des Manuskript-Standards auf 5.25"-Floppy-Disk, Association of American Publishers, Electronic Manuscript Series, August 1987

Alle diese Dokumente sind erhältlich bei:

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

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

Fig. 3: Referenz-Einheiten für jedes europäische Zeichen

Das Wesen des SGML-Standard wäre verfehlt, wenn sich nun der deutsche PC-Besitzer 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 und „Ä“ am Schirm anzuzeigen und „Ä“ auf dem Drucker auszudrucken, wenn der Benützer „Ä“ auf der Tastatur eingegeben hat. Sollte dem Autor jedoch das „Ä“ auf der Tastatur fehlen, wird er allerdings gerne zu „„Ä“ Zuflucht nehmen. Für griechische und kyrillische Buchstaben hat man denn auch neben den Einheiten-Referenzen (*α β γ ...*) auch eine Transliteration definiert, die zwischen Beginn- und End-Marken des entsprechenden Alphabets stehen. Um zusammenhängende Texte in diesen Sprachen zu schreiben, ist diese sehr viel praktischer als die Einheiten-Referenzen.

Name	*Character*	*Keyed*	*Character*	*Keyed*
Alpha	α	a	Α	A
Beta	β	b	Β	B
Gamma	γ	g	Γ	G
Delta	δ	d	Δ	D
Partial	∂	6		
Del (nabla)			∇	1
Epsilon	ϵ	e	Ε	E
(Cursive)	ε	3		
Digamma	ϝ	5		
Zeta	ζ	z	Ζ	Z
Eta	η	h	Η	H
Theta	θ	q	Θ	Q
(Cursive)	ϑ	j	Θ	J
Iota	ι	i	Ι	I
Kappa	κ	k	Κ	K
(Cursive)	ϰ	8		
Lambda	λ	l	Λ	L
Mu	μ	m	Μ	M
Nu	ν	n	Ν	N
Xi	ξ	x	Ξ	X
Omicron	ο	o	Ο	O
Pi	π	p	Π	P
	ϖ	2		
Rho	ρ	r	Ρ	R
(Cursive)	ϱ	9		
Sigma	σ	s	Σ	S
(Final)	ς	v		
Tau	τ	t	Τ	T
Upsilon	υ	u	Υ	U
Phi	ϕ	f	Φ	F
(Cursive)	φ	4		
Chi	χ	c	Χ	C
Psi	ψ	y	Ψ	Y
Omega	ω	w	Ω	W

Fig. 4: ASCII-Transliteration von Griechisch/Mathematisch

Als Referenzen im Sinne von „Verweisen“ werden auch Bezüge auf externes Material bezeichnet: Verweise auf andere Texte, auf Bild-Dateien, Ton-Dateien. In einem gewissen Sinn sind ja auch die Einheiten-Referenzen Verweise auf externe (Buchstaben-) Bilder.

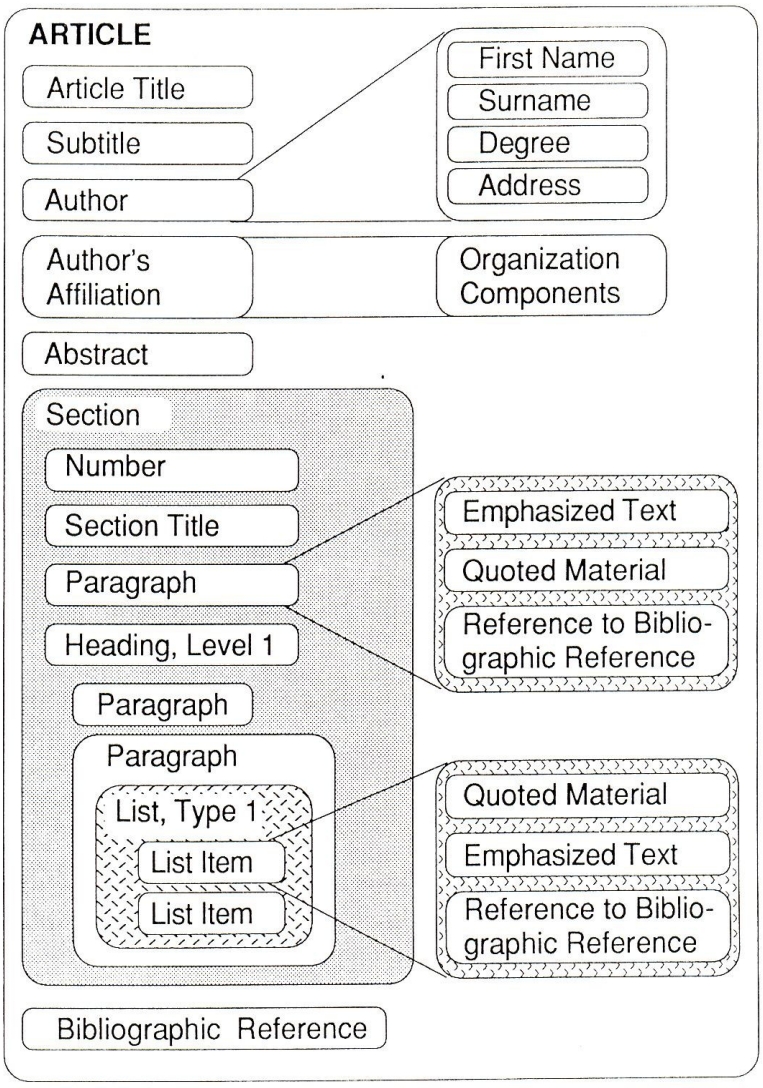
Neben den Referenzen braucht der Autor nur noch die *Textstruktur-Marken* zu kennen. Diese sind alle zwischen spitzen Klammern anzugeben. Jeder Textstruktur-Bestandteil hat einen Namen, der in der SGML-Dokument-Typ-Deklaration festgelegt ist. So steht etwa „ti“ für Titel. Der Beginn der Struktur ist mit dem Strukturnamen in eckigen Klammern anzugeben: *<ti>* für „hier beginnt der Titel“. Das Ende der Struktur ist durch spitze Klammern um einen Schrägstrich vor dem Strukturnamen gekennzeichnet. *</ti>* bezeichnet das Ende des Titels. 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 künstliche 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

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 einern 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, Affiliation, Adressen u.s.w. aufgeschlüsselt sein. Der eigentliche Text besteht aus einer Folge von einem oder mehr Abschnitten („section"). Ein Abschnitt besteht aus Abschnittnummer (evtl.), Abschnitttitel, einer Reihe von Absätzen, einem Untertitel der ersten Stufe (evtl.), gefolgt von einer Reihe von Absätzen ...

Obwohl sich das alles ein bisschen kompliziert anhört, zeigt ein Beispiel, dass ein sauber markierter Artikel recht lesbar ist. Graphische Übersichten über die Struktur der drei definierten Texttypen vereinfachen die Arbeit des Autors zusätzlich. Das praxisorientierte Autorenhandbuch von EPSIG hat denn auch nur fünfzig Seiten, wovon mehr als die Hälfte für die Liste der Spezialzeichen, den IRV-Zeichensatz und Beispiele verwendet ist. Überall im Text kann man Kommentare einfügen, die nicht zum Text zählen. Diese sind mit „<!--" und „-->" einzurahmen: *<!-- Dies ist ein Kommentar -->*.



Relationship for Basic Article Elements

Fig 5: Hierarchische Artikelstruktur (Übersicht)

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 (<ttg>) (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

24

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.

25

Fig. 6: Hierarchische Artikelstruktur (Struktur-Definition)

Und wie sieht das mit meinen Texten auf dem PC aus?

Solange die heute verbreiteten grösseren Textverarbeitungs-Systeme SGML noch nicht offiziell unterstützen, kann man die Austauschbarkeit von Texten maximieren, wenn man Leerzeilen zwischen den Absätzen hat, wenig „Firlefanz“ des Textsystems verwendet, statt Gedankenstriche „--“ eintippt und für Spezialzeichen, die nicht auf der Tastatur vorkommen, die offizielle Referenz-Einheit als Bezeichnung verwendet. Silbentrennung sollte grundsätzlich unterbleiben und dem Textsystem überlassen werden. Die Buchstaben „&“ und „<“, welche in SGML Steuerfunktionen haben, sollten als „&“ (= ampersand[15]) und „<“ (= less than) geschrieben werden. Vor Leerschlägen, wo keine Verwechslung zu befürchten ist, sind diese Umschreibungen unnötig. Einheitliche Markierung gleichartiger Textteile ist sogar dann schon nützlich, wenn es sich nicht um die offiziellen SGML-Markierungen handelt.

Als Leitfaden für die offiziellen Referenz-Namen und Dokument-Strukturen ist der oben erwähnte schlanke „Author's Guide“ von EPSIG eine unschätzbare Hilfe. Als Nachschlagewerk steht der vollständige „Reference Manual“ zur Verfügung. Die PC-Diskette „AAP Document Type Definitions and Short References“ enthält alle Deklarationen der Standard-Dokumente.

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.

Der Druck auf die Hersteller von Textsystemen wird in Kürze recht gross werden, so dass in den meisten Systemen neben dem ASCII-Export auch eine SGML-Schnittstelle vorliegen wird. Diese Umstellung wird

15 Aus „and-per-se and“ für das Zeichen für das lateinische „et“.

aufgrund der Logik der Benutzerschnittstelle je nach Textsystem mehr oder weniger Mühe bereiten. Schon heute gibt es eine Reihe von SGML-verarbeitenden Programmen[16]. Selbstverständlich kann einem Autor nur zugemutet werden, SGML-Text zu produzieren, wenn er mindestens ein System hat, dass seine Syntax prüft. Noch besser wird ein Textsystem sein, das die SGML-konforme Speicherung selber übernimmt.

Da im Zusammenhang mit Textdatenbanken, CD-ROM-Publikationen und in Kommunikation mit grossen amerikanischen Verlegern der neue Standard bald unumgänglich werden wird, braucht der deutsche Autor heute wohl nur wenige Monate zu warten, bis entsprechende Programme auf dem Markt sind. Verwendung von SGML bedeutet keinen Verzicht auf WYSIWYG: Auch wenn die Datenstrukturen konvergieren, wird die Vielfalt der Schnittstellen bleiben.

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. Mit SGML erhält die Branche einen seit langem dringend benötigten Standard.

Was ist die Zukunft von SGML?

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.

An der CD-ROM Expo in Washington (3./4. Okt. 1989) ist SGML schon eines der acht „in-depth tutorials“ für CD-ROM-Verleger und kommt noch in einem weiteren Konferenzbeitrag zum Zug.

Für die kluge Speicherung, Recherche und Weiterverarbeitung von Texten ist der SGML-Standard ein Segen. Dank SGML wird die Integration von Texten in einer grossen Datenbank kaum mehr Aufwand erfordern. Dank hoher 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 vorn Buch in eine Bibliothek.

Momentan werden von verschiedensten Interessengruppen neue Dokumenttypen in SGML definiert (Dissertation, Technisches Handbuch. . .). Auch diese Bewegung lässt uns schliessen, dass es von SGML kein Zurück mehr gibt.

Sollten die heute propagierten Dokumenttypen in der Anwendung wirklich Verbreitung finden, dürften alle textbasierten Datenbanksysteme die SGML-Struktur in Zukunft zentral verwenden. In den meisten Fällen sind wohl textbasierte flexible Retrievalsysteme schon heute angemessener als die verbreiteten dBASE-Dinosaurier.

Wohin führen die standardisierten Texte?

Bei Betrachtung des unaufhaltsam scheinenden Vormarschs der SGML-Textstandards fragt man sich unwillkürlich, welche Texte aufgrund dieser Standards ausgeschlossen werden. Welche werden favorisiert? Beeinflusst die standardisierte Struktur die möglichen Inhalte?

Der SGML-Kommission kann behutsamer Umgang mit Texten bescheinigt werden. Dank der Festlegung der Struktur der Strukturbeschreibung scheint sie die Möglichkeiten des literarischen Ausdrucks weitgehend zu respektieren. Dennoch: ein Autor, dessen Text eine neu zu definierende Struktur hat, wird kaum in der

---

16 Unter anderen:

ArborText Inc., *The Publisher 2.0*, auf SUN-Workstations, Ann Arbor, Michigan, U S A,

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

Sobemap, *MARK-IT*, in C auf UNIX oder MS-DOS (eingebaut in Texet, Interleaf, Scribe, XyVision), Brüssel, Belgien, 1988.

Lage sein, diese in SGML zu „programmieren“. Texte, die bisher ohne explizite Struktur auskamen, werden zur Deklaration getrieben. Dabei ist neben einem eventuellen Verlust an struktureller Vielfalt ein Gewinn an Qualität nicht auszusch1iessen.

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 benützen 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.

Hartwig Thomas, 8.11.89