

# Single Source Publishing

Modularer Content für ePub & Co.



Sissi Closs

# Single Source Publishing

Modularer Content für EPUB & Co.

Sissi Closs  
Single Source Publishing  
ISBN: 978-3-86802-265-0

© 2011 entwickler.press  
Ein Imprint der Software & Support Media GmbH

Bibliografische Information Der Deutschen Bibliothek  
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen  
Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über  
<http://dnb.ddb.de> abrufbar.

Ihr Kontakt zum Verlag und Lektorat:  
Software & Support Media GmbH  
entwickler.press  
Geleitsstr. 14  
60599 Frankfurt am Main  
Tel.: +49 (0)69 630089-0  
Fax: +49 (0)69 930089-89  
[lektorat@entwickler-press.de](mailto:lektorat@entwickler-press.de)  
<http://www.entwickler-press.de>

Lektorat: Sebastian Burkart  
Korrektorat: Frauke Pesch  
Satz: Pobporn Fischer  
Belichtung, Druck & Bindung: M.P. Media-Print Informationstechnologie GmbH, Paderborn

Alle Rechte, auch für Übersetzungen, sind vorbehalten. Reproduktion jeglicher Art (Fotokopie, Nachdruck, Mikrofilm, Erfassung auf elektronischen Datenträgern oder anderen Verfahren) nur mit schriftlicher Genehmigung des Verlags. Jegliche Haftung für die Richtigkeit des gesamten Werks kann, trotz sorgfältiger Prüfung durch Autor und Verlag, nicht übernommen werden. Die im Buch genannten Produkte, Warenzeichen und Firmennamen sind in der Regel durch deren Inhaber geschützt.

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>13</b>
<b>1 Ursprung und Hintergrund</b>	<b>15</b>
1.1 Markup-Sprachen: die bescheidenen Anfänge	15
1.2 DTP: die erste Single-Source-Publishing-Welle	18
1.2.1 Die Gestaltung kostet viel	19
1.2.2 Ein Tool macht abhängig	20
1.2.3 Jedes Tool hat seine Grenzen	20
1.3 Back to the Roots	21
1.3.1 Metasprache lässt Freiräume	21
1.3.2 Standardisierung gibt Sicherheit	23
1.3.3 Trennung von Inhalt und Darstellung macht flexibel	23
1.3.4 Formatierung ist nicht alles	26
1.4 Die letzte Stufe: das Baukastenprinzip	27
1.5 Irrungen und Wirrungen über Single Source Publishing	28
1.6 Das Wichtigste aus Kapitel 1	28
<b>2 Der jüngste Trend – Mobil</b>	<b>31</b>
2.1 Gerätevielfalt	31
2.1.1 Smartphone	32
2.1.2 E-Book-Lesegerät	34
2.1.3 Tablet-PC	35
2.2 Was ist das Besondere an der mobilen Welt?	36
2.2.1 Wir verlassen uns auf unseren ständigen Begleiter	37
2.2.2 Wir nutzen viele Sinne	37
2.2.3 Wir nehmen Informationen anders wahr	39
2.2.4 Wir müssen nicht suchen	39
2.2.5 Wir erhalten kontextbezogene Inhalte	40
2.2.6 Beteiligung ist möglich	42

2.3	Elektronische Bücher	43
2.3.1	Was ist EPUB?	43
2.3.2	Geschichte	43
2.3.3	Bestandteile des EPUB-Standards	43
2.3.4	Bestandteile eines EPUB-Dokuments	44
2.3.5	Besonderheiten von EPUB	44
2.3.6	Unterschied zwischen PDF und EPUB	45
2.4	Erstellung von Inhalten für mobile Geräte	45
2.4.1	Topic-Struktur und DITA als gesunde Basis	46
2.4.2	Publikationsprozess für mobile Doku	46
2.5	Irrungen und Wirrungen über mobile Doku	48
2.6	Das Wichtigste aus Kapitel 2	48
<b>3</b>	<b>Einstieg in die Topicorientierte Strukturierung</b>	<b>49</b>
3.1	Was ist ein Topic?	49
3.2	Was sind Beziehungen zwischen Topics?	50
3.3	Was ist ein Informationsgebilde?	51
3.4	Welche Vorteile hat die Topic-orientierte Strukturierung?	51
3.5	Weitere Beispiele	55
3.6	Irrungen und Wirrungen über Topic-orientierte Strukturierung	56
3.7	Das Wichtigste aus Kapitel 3	56
<b>4</b>	<b>Topics</b>	<b>57</b>
4.1	Was kennzeichnet ein Topic?	57
4.2	Wie sind Topics entstanden?	58
4.3	Darstellung von Topics	59
4.3.1	Darstellung von Topics bei der Konzeption	59
4.3.2	Darstellung von Topics bei der Erstellung	59
4.3.3	Darstellung von Topics bei der Nutzung	60
4.4	Vorteile von Topics	62
4.5	Irrungen und Wirrungen über Topics	62
4.6	Das Wichtigste aus Kapitel 4	62

<b>5</b>	<b>Beziehungen</b>	<b>63</b>
5.1	Welche Beziehungen gibt es?	63
5.1.1	Aus der Papierwelt bekannte Beziehungen	63
5.1.2	Weitere Arten von Beziehungen in der elektronischen Welt	64
5.2	Merkmale von Beziehungen	67
5.2.1	Kardinalität	67
5.2.2	Richtung	68
5.3	Darstellung von Beziehungen	69
5.3.1	Darstellung von Beziehungen bei der Konzeption	69
5.3.2	Darstellung von Beziehungen bei der Erstellung	70
5.3.3	Darstellung von Beziehungen bei der Nutzung	71
5.4	Links	71
5.4.1	Wie sind Links entstanden?	72
5.4.2	Darstellung von Links auf der Erstellungsseite	73
5.4.3	Darstellung von Links auf der Nutzungsseite	74
5.4.4	Bekannte Problempunkte	80
5.5	Irrungen und Wirrungen über Beziehungen und Links	81
5.6	Das Wichtigste aus Kapitel 5	81
<b>6</b>	<b>Informationsgebilde</b>	<b>83</b>
6.1	Was ist ein Informationsgebilde?	83
6.2	Gedrucktes Dokument	84
6.2.1	Zusammengehörigkeit	84
6.2.2	Organisation	85
6.2.3	Orientierungs- und Zugangshilfen	86
6.3	Hyperdokument	90
6.3.1	Hypermedia-Systeme	91
6.3.2	Wie entsteht ein Hyperdokument?	92
6.3.3	Wie wird ein Hyperdokument dargestellt?	94
6.4	Was ist ein Topic-basiertes Informationsgebilde?	99
6.4.1	Topic-Struktur wenig betont, aber vorhanden	99
6.4.2	Topic-Struktur mehr betont, aber nicht immer vorhanden	100
6.4.3	Topic-Struktur für Single-Source-Umgebungen	102

6.5	Besonderheiten von Topic-basierten Informationsgebilden	103
6.5.1	Orientierungshilfen	103
6.5.2	Wiederverwendung	104
6.5.3	Schneller Zugang	106
6.6	Darstellung von Topic-basierten Informationsgebilden	108
6.6.1	Darstellung von Topic-basierten Informationsgebilden bei der Konzeption	109
6.6.2	Darstellung von Topic-basierten Informationsgebilden bei der Erstellung	110
6.6.3	Darstellung von Topic-basierten Informationsgebilden bei der Nutzung	112
6.7	Irrungen und Wirrungen über Informationsgebilde	112
6.8	Das Wichtigste aus Kapitel 6	113
<b>7</b>	<b>Modellieren und Standardisieren</b>	<b>115</b>
7.1	Modelle	115
7.1.1	Modelle für Informationsgebilde	115
7.1.2	Bisherige Modelle reichen nicht aus	116
7.1.3	Adäquate Modelle werden benötigt	117
7.2	Das Beispiel vom Haushalt	118
7.3	Standardisierung	119
7.3.1	Warum Standardisierung?	119
7.3.2	Welche Aufgaben fallen bei der Standardisierung an?	120
7.3.3	Worin besteht das Risiko?	121
7.4	Klassifizierung	122
7.4.1	Was ist eine Klasse?	122
7.4.2	Wozu dienen Klassen?	122
7.4.3	Wie entsteht eine Klasse?	122
7.5	Irrungen und Wirrungen über Modelle und Standards	123
7.6	Das Wichtigste aus Kapitel 7	123
<b>8</b>	<b>Klassenkonzepttechnik</b>	<b>125</b>
8.1	Was ist die Klassenkonzepttechnik?	125
8.1.1	Minimalistischer Ansatz	125
8.1.2	Klassenbildung	126
8.1.3	Vorgehensmodell	127
8.1.4	Entstehung der Klassenkonzepttechnik	131



8.2	Topic-Klasse	133
8.2.1	Darstellung einer Topic-Klasse	133
8.2.2	Standards für Topic-Klassen	134
8.2.3	Modularisierungstechnik	136
8.2.4	Entwicklung eines Klassenkonzepts für Topics	140
8.3	Beziehungsklasse	146
8.3.1	Darstellung einer Beziehungsklasse	146
8.3.2	Standards für Beziehungsklassen	148
8.3.3	Grundregeln für die Festlegung von Beziehungen	150
8.3.4	Verfahren für die Ermittlung der Beziehungsklassen	151
8.4	Informationsgebildeklasse	156
8.4.1	Darstellung einer Informationsgebildeklasse	156
8.4.2	Standards für Informationsgebildeklassen	156
8.5	Vorteile des Klassenkonzepts	158
8.5.1	Vorteile für die Erstellungsseite	158
8.5.2	Vorteile für die Nutzungsseite	160
8.5.3	Keine Standardisierung um jeden Preis	160
8.6	Irrungen und Wirrungen über das Klassenkonzept	161
8.7	Das Wichtigste aus Kapitel 8	161
<b>9</b>	<b>Topic-orientiertes Single Sourcing einführen</b>	<b>163</b>
9.1	Mögliche Stolpersteine	163
9.1.1	Stolperstein Bereichsauswahl	163
9.1.2	Stolperstein Technik	164
9.1.3	Stolperstein Arbeitsorganisation	166
9.2	Im vertrauten Umfeld starten	166
9.2.1	Strukturierung trainieren	167
9.2.2	Umsetzungsmöglichkeiten kennen lernen	172
9.2.3	Zusammenarbeit stärken	173
9.3	Organisation vorbereiten	175
9.4	Bereiche ausdehnen	176
9.5	Passende technische Umsetzung wählen	177
9.6	Irrungen und Wirrungen über die Einführung von Topic-orientiertem Single Sourcing	179
9.7	Das Wichtigste aus Kapitel 9	179

<b>10 DITA</b>	<b>181</b>
10.1 Was ist DITA?	181
10.2 Wie ist DITA entstanden?	182
10.3 Was steht für DITA zur Verfügung?	182
10.4 Was ist ein DITA-Topic?	183
10.5 Topics zusammenbauen	185
10.5.1 Verschachtelung von Topics	185
10.5.2 DITA-Map	185
10.6 Welche Möglichkeiten bietet DITA für die Verlinkung?	188
10.6.1 Link im Topic	188
10.6.2 Verlinkung über die Map	188
10.7 Adressierung	188
10.7.1 Direkte Adressierung	188
10.7.2 Mehr Freiheit durch indirekte Adressierung	188
10.8 Varianten über Variablen	189
10.9 Inhalte einbetten	189
10.9.1 Flexible Wiederverwendung über conref	189
10.9.2 Nicht-DITA-Texte einbetten	191
10.10 Zusammenarbeit in geregelten Bahnen	192
10.11 Wie entsteht ein Informationsgebilde?	193
10.12 DITA-Spezialisierung	195
10.13 Warum lohnt es sich, auf DITA zu setzen?	196
10.14 Irrungen und Wirrungen über DITA	197
10.15 Das Wichtigste aus Kapitel 10	198
<b>11 DITA-Kurzreferenz</b>	<b>199</b>
11.1 Inhaltliche Zusammenstellung der DITA-Elemente	199
11.1.1 Allgemeine Topic-Elemente	199
11.1.2 Elemente für Konzept-Topics	200
11.1.3 Elemente für Referenz-Topics	200
11.1.4 Elemente für Aufgaben-Topics	201
11.1.5 Elemente für den Topic-Rumpf	202
11.1.6 Elemente für Tabellen	203
11.1.7 Elemente für typografische Auszeichnungen	204
11.1.8 Domänenspezifische Elemente	204
11.1.9 Elemente für Metadaten	208

11.1.10	Elemente für Links auf verwandte Topics	209
11.1.11	Elemente für die Spezialisierung	210
11.1.12	Elemente für DITA-Maps	210
11.1.13	Elemente für Buch-Maps	211
11.1.14	Elemente für Glossare	214
11.1.15	Elemente für den Index	214
11.1.16	Elemente für dynamisches Einbetten	215
11.1.17	Elemente für Adressangaben	215
11.1.18	Elemente für Learning and Training	216
11.1.19	Elemente für Sicherheitshinweise	221
11.1.20	Elemente für Arbeitsanleitungen	222
11.1.21	Elemente für Klassifikationen	223
11.2	Alphabetische Referenz	224
<b>12</b>	<b>Literaturverzeichnis</b>	<b>323</b>
12.1	Bücher und Zeitschriftenartikel	323
12.1.1	Standardwerke für Technische Dokumentation	323
12.1.2	Content Management	325
12.1.3	Metadaten	328
12.1.4	Strukturierung und Single Sourcing	328
12.1.5	XML	329
12.1.6	DITA	330
12.1.7	Mobile Doku	330
12.2	Internetquellen	331
12.3	Internetadressen für DITA	332
	<b>Stichwortverzeichnis</b>	<b>333</b>



# Vorwort

**Dauerten wir unendlich,  
so wandelte sich alles.  
Da wir aber endlich sind,  
bleibt vieles beim alten.**

(Bertolt Brecht, um 1955)

Wissen gehört zu den wichtigsten Ressourcen unserer Zeit. Um es gezielt und schnell aufbauen zu können, benötigen wir ein Informationsmanagement, das die effiziente Erstellung, Verwaltung und Nutzung von Informationen ermöglicht.

Die zunehmende Digitalisierung der Inhalte stellt Informationsanbietende und -nutzende vor die Herausforderung, immer größere Informationsberge von wachsender Vielfalt in immer kürzerer Zeit verarbeiten zu müssen. Längst wurde erkannt, dass Programmsysteme allein nicht ausreichen, um diese Herausforderung zu meistern, sondern auch passende Konzepte, Methoden und Vorgehensweisen benötigt werden. Eine Kernidee zur Lösung wird heute als „Single Sourcing“ bezeichnet. Sie zielt darauf ab, nur eine Quelle zu verwalten und Vorhandenes so weit wie möglich wieder zu verwenden, um den Aufwand auf der Erstellungsseite zu reduzieren und gleichzeitig die Qualität für die Nutzung zu verbessern. Mit den mobilen Medien bekommt Single Sourcing eine weitere Dimension und ist gefragter denn je. Dieses Buch beschreibt die Konzepte und Methoden für Single Sourcing und stellt mit dem Klassenkonzept eine Technik vor, mit der effizient und in Stufen ein stabiles Gerüst für Single-Source-Umgebungen beliebiger Größe und Komplexität erstellt werden kann. Die Klassenkonzepttechnik wird seit über zwanzig Jahren erfolgreich in der Praxis eingesetzt. In diesem Buch ist sie ausführlich beschrieben. Es geht auch darum zu zeigen, dass die Konzepte nicht neu sind, sondern größtenteils schon existieren, seit es Dokumente gibt und insbesondere, seit Dokumente elektronisch erstellt werden. Viele Mehrfachaufwände und Irrwege könnten vermieden werden, wenn ganz im Sinne von Single Sourcing die Wiederverwendung bewährter Konzepte explizit und gezielt betrieben würde, statt sie unter anderem Namen immer wieder neu zu erfinden.

Schließlich möchte das Buch herausstellen, dass Wissen trotz aller Systematik und Technik nicht auf Knopfdruck aus Computern abrufbar ist, sondern untrennbar zu Menschen gehört und zu deren Fähigkeit und Bereitschaft, mit Informationen kooperativ und konstruktiv umzugehen.

Die zweite Auflage enthält ein neues Kapitel zum Thema mobile Dokumentation und berücksichtigt den aktuellen DITA-Standard in der Version 1.2.

### **Danksagung**

Viele Personen haben auch an dieser 2. Auflage meines Buchs mitgewirkt. Ich bedanke mich bei allen, die mir geholfen haben, bei meinen MitarbeiterInnen meiner Firmen Comet Computer und Comet Communication sowie bei meinen Kollegen und den Studierenden im Studiengang Technische Redaktion an der Hochschule Karlsruhe.

Mein ganz besonderer Dank gilt allen, die direkt an der Erstellung des Buchs beteiligt waren: allen voran Markus Abt, der zum Gelingen entscheidend beigetragen hat; Carmen Krug und Eva Fitz für ihre Mitarbeit an der DITA-Kurzreferenz sowie Tina Wacek und Franziska Bechthold für ihre Unterstützung bei der Recherche und der grafischen Gestaltung.

Vielen Dank an meinen Lektor, Sebastian Burkart, und den entwickler.press-Verlag für die verständnisvolle, flexible und freundliche Zusammenarbeit. Vielen Dank Thomas Zschocke für die wertvollen fachlichen Hinweise und Ergänzungen.

Und schließlich ein großes Dankeschön an meine Familie.

# 1

## Ursprung und Hintergrund

Auch wenn der Begriff Single Source Publishing erst Ende der 1990er Jahre geprägt wurde, gehen die Ursprünge weit zurück bis zu den Anfängen des Computers. Im Kern geht es immer darum, Informationen effizient elektronisch zu erfassen, zu speichern und sie dann für Menschen und Programme wieder nutzbar zu machen.

Getrieben wird die Entwicklung von der Dokumentation und der Technik. Auf Dokumentationsseite stehen Qualitäts- und Kostenaspekte im Vordergrund, auf technischer Seite spielen neben immer neuen Hard- und Softwareentwicklungen die Formate eine zentrale Rolle.

### 1.1 Markup-Sprachen: die bescheidenen Anfänge

Angefangen hat es damit, dass aus elektronisch gespeicherten Textinformationen formatierte Papierausdrucke in einem für Menschen geeigneten Layout produziert werden sollten. Drucker mit Formatierungsfunktionen konnten diese Umsetzung leisten. Unglücklicherweise hatte aber jeder Drucker seine eigenen Druckbefehle. Wollte man eine Formatierungsfunktion, zum Beispiel Fettdruck, nutzen, musste der richtige Befehl für den gewählten Drucker in den Text geschrieben werden.

Es dauerte nicht lange, bis erste technische Verbesserungen entwickelt wurden. Für Drucker wurden generische Sprachen definiert. Postscript und PDF von Adobe sowie PCL von Hewlett Packard haben sich als Industriestandards etabliert.

Für die Autoren entstanden Auszeichnungssprachen (Markup-Sprachen). Diese ermöglichen eine funktionale und für den Menschen leicht handhabbare Eingabe der Layoutbefehle. Spezielle Satzprogramme erzeugen aus den Befehlen der Markup-Sprache das für einen Drucker benötigte Ausgabeformat. Die Aufgabe, für jeden Drucker ein spezifisches Dokument zu produzieren, wurde also vom Menschen auf Programme verlagert.

Jede große Firma entwickelte ihre eigene Markup-Sprache und ihre eigenen Satzprogramme.

### Beispiel

Die Auszeichnungssprache von Siemens hieß ICP, das zugehörige Satzprogramm COSY.

ICP-Befehl	Bedeutung
.1	Leerzeile
.N	Zeile mit Schriftbefehlen
.W	Stichwort
Y .. y	Zeilen- und stellengleiche Wiedergabe
#D .. #d	Dicktengleiche Schriftart
#H .. #h	Fettschrift
#K .. #k	Kursiv
#ZBI	Bildschirmsymbol

Abbildung 1.1: ICP-Befehle

Die Autoren mussten die Formatierungsbefehle und die Umlautkodierung (^U statt Ü, ^s statt ß usw.) kennen und in den Text schreiben. Abbildung 1.1 zeigt einen mit ICP ausgezeichneten Textabschnitt.

```
3 ^Uberschrift
.1
. . Punktliste
.1
. . Punktliste
.1
  Text mit #Kkursiver#k und #Hhalbfetter#h Auszeichnung.
  Die deutschen Umlaute wurden durch ein vorangestelltes ^ vor
  dem Buchstaben gekennzeichnet.
.1
```

Abbildung 1.1: Auszug aus ICP-Datei

Auch Grafik war schon möglich. Einfache, aus Bindestrichen und senkrechten Strichen zusammengesetzte Grafiken konnten (sicherlich etwas mühsam) erstellt werden, wurden aber wie die Formatierung erst nach der Konvertierung durch das Satzprogramm als Grafik dargestellt.



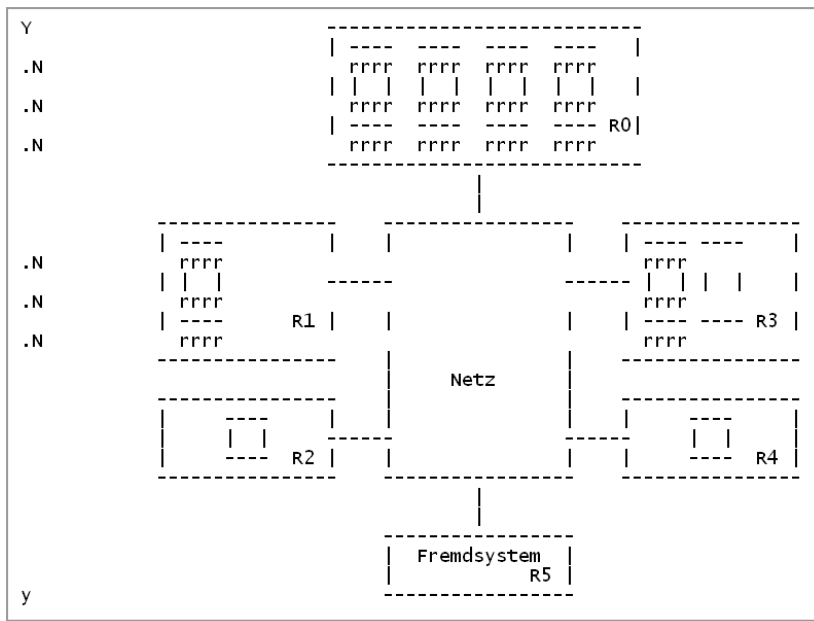


Abbildung 1.2: ICP-Grafik bei der Erfassung

Nach dem Satzprogrammdurchlauf entstand die druckfertige Grafik:

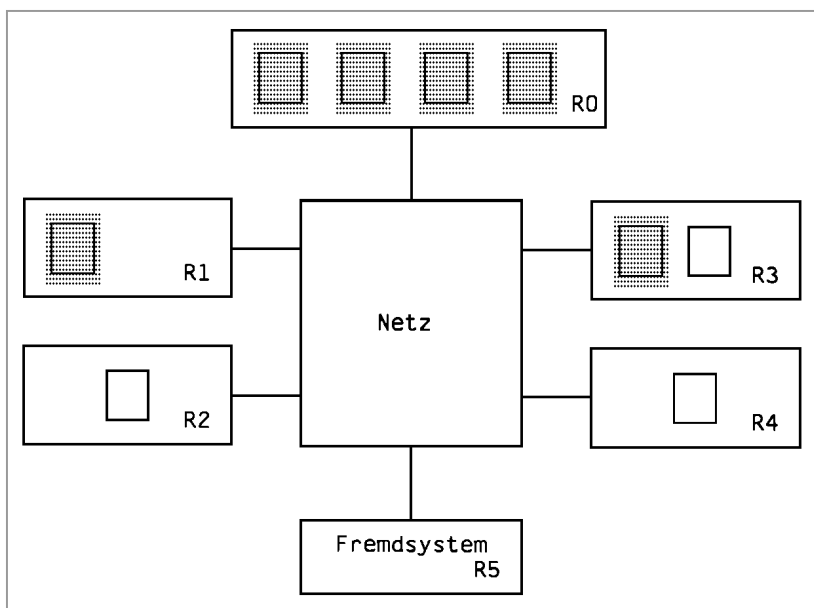


Abbildung 1.3: ICP-Grafik gedruckt

Trotz der etwas mühsamen Texterfassung hat dieses Verfahren durchaus Vorteile:

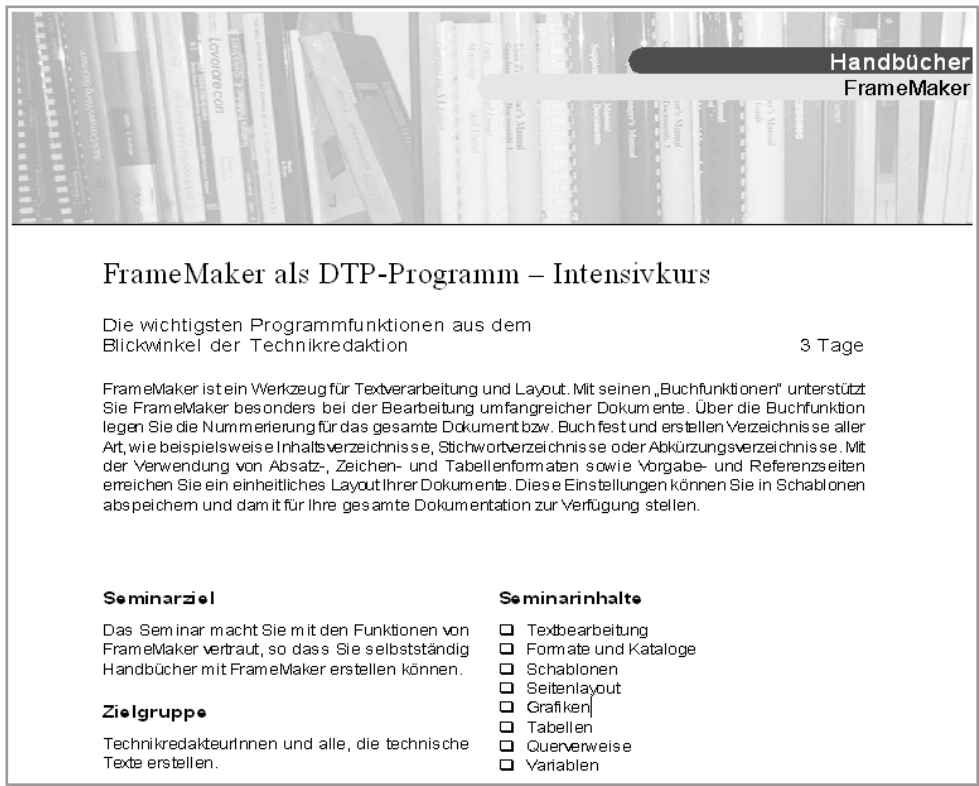
- Text, Formatierungsangaben und sogar Grafik werden in derselben Quelle erfasst.
- Die Quelle ist eine Textdatei, die Menschen und Programme lesen können.
- Die Dateien sind klein. Selbst große Dokumente mit mehreren hundert Seiten können in einer Datei abgelegt und bearbeitet werden. Der komplette Inhalt kann in einer Quelle (Single Source) verwaltet werden.
- Einige Aufgaben wie Suchen und Ersetzen sind auf dieser Basis konkurrenzlos schnell durchführbar. Sind die Befehle bekannt, ist auch die Erfassung schnell, weil Formatierungsbefehle direkt mit dem Text eingetippt werden, ohne den Arbeitsfluss durch Mausbedienung und Menüauswahl unterbrechen zu müssen.
- Die Formatierung ist zentral durch das Satzprogramm festgelegt. Das Layout ist auf Autorensseite nicht beeinflussbar. Das sichert die Konsistenz.

Für die Technische Dokumentation reicht in der Regel ein einfaches, funktionales Layout. Somit war mit Markup-Sprachen eine Produktion möglich, die schon viele Single-Source-Aspekte verwirklichte, aber leider noch nicht perfekt war:

- Das Produktionsverfahren war umständlich und langsam. Die Ausdrücke wurden im Batch-Verfahren über Nacht produziert. Und so blieb es nicht aus, dass ab und zu noch schnell letzte Änderungen oder Feinschliffe von der Satzstelle direkt in die Ausgabe eingebracht wurden und die Quelle mit der Ausgabe nicht mehr übereinstimmte.
- Die Gestaltungsmöglichkeiten waren bescheiden. Hochwertige Bilder und anspruchsvolles Layout konnten bis Ende der 1980er Jahre nur mit der traditionellen manuellen Satz- und Drucktechnik erstellt werden.
- Jede Firma hatte ihre eigene Sprache. Ein Austausch der Dokumente war nur mit hohem manuellen Aufwand oder eigens programmierten Konvertern möglich. Erst nach und nach entstanden firmenübergreifende Auszeichnungssprachen und zugehörige Satzprogramme. Die prominentesten sind nroff/troff aus der UNIX-Welt und TeX von Donald Knuth. TeX und das darauf basierende LaTeX haben heute noch viele treue Anhänger.

## 1.2 DTP: die erste Single-Source-Publishing-Welle

Ende der 1980er Jahre war es dann soweit: Die Entwicklung von DTP-(Desktop-Publishing-)Systemen löste eine Revolution in der Dokumentationswelt aus. Ein einziges Programm ersetzt eine Kette von Tools und manuellen Tätigkeiten. Inhalte können schon beim Erstellen formatiert werden. Aus der Quelle können anspruchsvolle Druckausgaben ohne zusätzliche Werkzeuge produziert werden. WYSIWYG (What You See Is What You Get) wurde zum geflügelten Wort. Bekannte Systeme wie Interleaf, FrameMaker, Ventura Publisher und QuarkXPress haben sich schnell verbreitet.



**Handbücher**  
**FrameMaker**

## FrameMaker als DTP-Programm – Intensivkurs

Die wichtigsten Programmfunktionen aus dem Blickwinkel der Technikredaktion 3 Tage

FrameMaker ist ein Werkzeug für Textverarbeitung und Layout. Mit seinen „Buchfunktionen“ unterstützt Sie FrameMaker besonders bei der Bearbeitung umfangreicher Dokumente. Über die Buchfunktion legen Sie die Nummerierung für das gesamte Dokument bzw. Buch fest und erstellen Verzeichnisse aller Art, wie beispielsweise Inhaltsverzeichnisse, Stichwortverzeichnisse oder Abkürzungsverzeichnisse. Mit der Verwendung von Absatz-, Zeichen- und Tabellenformaten sowie Vorgabe- und Referenzseiten erreichen Sie ein einheitliches Layout Ihrer Dokumente. Diese Einstellungen können Sie in Schablonen abspeichern und damit für Ihre gesamte Dokumentation zur Verfügung stellen.

<b>Seminarziel</b>	<b>Seminarinhalte</b>
Das Seminar macht Sie mit den Funktionen von FrameMaker vertraut, so dass Sie selbstständig Handbücher mit FrameMaker erstellen können.	<input type="checkbox"/> Textbearbeitung
<b>Zielgruppe</b>	<input type="checkbox"/> Formate und Kataloge
Technikredakteurinnen und alle, die technische Texte erstellen.	<input type="checkbox"/> Schablonen
	<input type="checkbox"/> Seitenlayout
	<input type="checkbox"/> Grafiken
	<input type="checkbox"/> Tabellen
	<input type="checkbox"/> Querverweise
	<input type="checkbox"/> Variablen

Abbildung 1.4: Inhalt mit Layout in einer FrameMaker-Quelle

Die Inhalte werden in derselben Quelle mit demselben Tool erfasst und gestaltet. Zusätzlich bietet ein DTP-Tool nützliche Funktionen, wie die Generierung von Inhalts- und Stichwortverzeichnissen, die zeitraubende und fehleranfällige Handarbeiten überflüssig machen.

Für die Produktion von Druckausgaben schien DTP das nahezu perfekte Verfahren zu sein. Aber es gab einige Wermutstropfen.

### 1.2.1 Die Gestaltung kostet viel

Die Autoren verbringen viel Zeit mit der Gestaltung und den Tools. Je mehr Möglichkeiten ein Werkzeug bietet, desto professioneller muss damit umgegangen werden. Spezielle Toolkenntnisse und -erfahrungen sind erforderlich, um es richtig zu bedienen. Für den eigentlichen Inhalt bleibt entsprechend weniger Zeit.

### 1.2.2 Ein Tool macht abhängig

Die Möglichkeit, schon beim Editieren zu formatieren, wird technisch durch toolspezifische, binäre Dateiformate teuer erkaufte. Nur die binären Formate ermöglichen eine schnelle WYSIWYG-Bearbeitung. Ein Mensch kann binäre Formate nicht lesen. Auch andere Anwendungen können die Daten nicht direkt übernehmen. Es werden Filter und Konverter benötigt, die in der Regel nicht verlustfrei arbeiten. Selbst bei der Weiterentwicklung desselben Tools ist damit zu rechnen, dass die formatierten Daten nicht problemlos ausgetauscht werden können.

Je mehr Inhalte bereits erstellt wurden, desto schmerzlicher wird diese Abhängigkeit. Bei einem irgendwann nicht mehr vermeidbaren Versions- oder Toolwechsel muss alles, was nicht programmgesteuert übernommen wird, mühsam neu gemacht werden.

### 1.2.3 Jedes Tool hat seine Grenzen

Auch wenn die Systeme immer ausgefeilter werden, hat jedes Tool seine Grenzen. Das bekam die Dokumentationswelt deutlich zu spüren, als neben dem Papier elektronische Medien mit immer neuen Möglichkeiten für die Publikation aufkamen. Für die Nutzerseite sollten die Möglichkeiten natürlich eingesetzt werden, aber die herkömmlichen DTP-Systeme konnten nur Ausgaben für den Druck produzieren. Folglich entstanden neue Werkzeuge und spezielle Formate für die neuen Darstellungsformen: Autorenwerkzeuge für Onlinehilfen und Multimediaanwendungen, Editoren für HTML-Seiten und Grafikwerkzeuge. Die einst als große Errungenschaft gefeierte Verschmelzung von Inhalt und Formatierung stellte sich jetzt als nachteilig heraus. Derselbe Inhalt musste aus technischen Gründen in mehreren Quellen mit unterschiedlichen Werkzeugen erstellt und gepflegt werden. Die bekannten Probleme redundanter Quellen blieben nicht aus und wurden immer größer, je schneller der Umfang wuchs. Fehler und Inkonsistenzen schlichen sich ein, weil die Aktualisierung nicht an allen Stellen gleichzeitig stattfinden konnte. Übersetzungsumfang und -aufwand stiegen erheblich.

Technisch wurde versucht, den Missstand durch Konverter zu lösen. Diese als Cross-Media-Publishing angebotenen Toolumgebungen ermöglichen die Erstellung des Inhalts in einer Quelle mit einem Werkzeug. Toolinterne oder eigenständige Konverter produzieren aus der Quelle die gewünschten anderen Ausgabeformate.

Das Redundanzproblem konnte auf diese Weise verringert werden, aber Cross-Media-Publishing bleibt immer nur ein Kompromiss. Eine "ver-online-te" Papierdokumentation ist noch lange keine optimale Onlinedokumentation, eine für die Druckausgabe konvertierte Onlinedokumentation ist keine perfekte Papierversion.

## 1.3 Back to the Roots

Die dargestellten Nachteile der auf Formatierung ausgerichteten Verfahren haben die rein textbasierte Produktionsweise mit Markup-Sprachen nie vergessen lassen. Parallel zur schillernden DTP- und WYSIWYG-Welt hat sich daher auch die Markup-Welt weiterentwickelt.

Ihre schlagenden Vorteile, nämlich reines Textformat und damit geringe Toolabhängigkeit sowie einfache Programmierung, wurden konsequent weiterverfolgt. Sie gipfelten in der Erfindung der Metasprachen SGML (Standard Generalized Markup Language) und XML (eXtensible Markup Language) mit weitreichenden Konsequenzen.

### 1.3.1 Metasprache lässt Freiräume

Die zündende Idee war, statt einer konkreten Markup-Sprache ein Regelwerk zu definieren, mit dem nach vorgegebenem Schema eigene Markup-Sprachen definiert werden können. Auf diese Weise wird dem Sprachenwildwuchs in gewissem Maß Einhalt geboten, aber die Individualität nicht ausgebremsst.

Eine SGML- oder XML-Markup-Sprache wird durch ein Vokabular und eine Grammatik definiert. SGML bzw. XML geben vor, wie das Vokabular und die Grammatikregeln gebildet, geschrieben und verwendet werden.

#### Beispiel

Die Grammatik für eine SGML- oder XML-Sprache kann als DTD (Dokument-Typ-Definition) angegeben werden. Abbildung 1.5 zeigt eine einfache DTD für Kochrezepte:

```
<!-- eine einfache Rezept-DTD -->

<!ELEMENT rezept (name, bild?, zutaten, zubereitung?) >
<!ELEMENT name (#PCDATA) >
<!ELEMENT bild EMPTY >
<!ATTLIST bild href CDATA #REQUIRED >
<!ELEMENT zutaten (teil+ | zutat+) >
<!ELEMENT teil (name, zutat+) >
<!ELEMENT zutat ((menge, einheit?)?, name) >
<!ELEMENT menge (#PCDATA) >
<!ELEMENT einheit (#PCDATA) >
<!ELEMENT zubereitung (schritt+) >
<!ELEMENT schritt (#PCDATA) >
```

Abbildung 1.5: XML-DTD für Kochrezepte

Wenn eine Sprache definiert ist, können Inhalte mit den Elementen der Sprache ausgezeichnet werden. Dazu werden die Inhalte mit einem Start- und Ende-Tag umrahmt, das den Elementnamen enthält. Für die Auszeichnung hat sich im SGML- und XML-Umfeld deshalb der Begriff „taggen“ etabliert.

### Beispiel

Mit der durch die DTD in Abbildung 1.5 definierten XML-Sprache kann ein Kochrezept wie folgt getaggt werden:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE rezept (View Source for full doctype...)>
- <rezept>
  <name>Donauwellen</name>
  <bild href="donauwellen.jpg" />
  - <zutaten>
    - <teil>
      <name>Boden</name>
      - <zutat>
        <menge>250</menge>
        <einheit>g</einheit>
        <name>Margarine</name>
      </zutat>
      - <zutat>
        <menge>200</menge>
        <einheit>g</einheit>
        <name>Zucker</name>
      </zutat>
      + <zutat>
      + <zutat>
      + <zutat>
      + <zutat>
    </teil>
    + <teil>
  </zutaten>
  - <zubereitung>
    <schritt>Alle Zutaten für den Teig gut verrühren.</schritt>
    <schritt>Auf ein gefettetes Backblech streichen und 2 Gläser Kirschen abtropfen lassen. Verteilen und bei 175 Grad 20-30 Minuten backen.</schritt>
    <schritt>Von 1 Liter Milch mit 2 Päckchen Puddingpulver und 1 El. Zucker Pudding kochen, auf den gebackenen Boden streichen.</schritt>
    <schritt>Butterkekse darauf verteilen und erkalten lassen.</schritt>
    <schritt>1/2 Pfund Puderzucker 2 Eier und 2 Esslöffel Kakao verrühren.</schritt>
    <schritt>1/2 Pfund Palmin erwärmen und tropfenweise unter die Masse geben.</schritt>
    <schritt>Creme auf die Kekse verteilen.</schritt>
    <schritt>Sehr fest werden lassen, am besten im Kühlschrank.</schritt>
  </zubereitung>
</rezept>
```

Abbildung 1.6: Getaggttes Rezept für Donauwellen

Dank der einheitlichen Formalismen ist jede Sprache nach dem gleichen Muster gestrickt. Deswegen können generische Programme geschrieben werden, die für jede Sprache einsetzbar sind. Ein Parser ist zum Beispiel ein solches, sehr zentrales Programm. Er prüft, ob die Sprache korrekt verwendet wird. Nicht jeder, der eine SGML- oder XML-Sprache definiert, muss seinen eigenen Parser programmieren. Vielmehr genügt ein Parser, der eine nach SGML- bzw. XML-Regeln erstellte Grammatik lesen kann und auf dieser Basis die korrekte Verwendung der Sprache prüft.

### **1.3.2 Standardisierung gibt Sicherheit**

SGML wurde 1986 als internationaler Standard ISO 8879 verabschiedet. Die bekannteste SGML-Sprache ist HTML. Trotz des guten Ansatzes hat SGML den großen Durchbruch nicht geschafft. Der Grund dafür ist, dass SGML teilweise zu komplex ist, was die Programmierung von allgemeinen SGML-Tools schwierig und teuer macht. Das WWW-Konsortium (W3C) definierte daher 1998 den neuen Standard XML unter dem Aspekt einfacher Handhabung und schneller Umsetzbarkeit.

Die Standardisierung schafft Neutralität und gibt mehr Sicherheit in Hinblick auf langfristige Verfügbarkeit.

### **1.3.3 Trennung von Inhalt und Darstellung macht flexibel**

Wie schon bei den ersten Markup-Sprachen wird mit SGML und XML die Inhaltsquelle von einer spezifischen Formatierung befreit, um dieselbe Quelle für unterschiedliche Darstellungen verwenden zu können. Die Formatierung kann weitgehend programmgesteuert erfolgen. Dazu wurden mit den Metasprachen auch Mechanismen für die Formatierung definiert, für deren Umsetzung generische Programme wie für die Syntaxprüfung geschrieben werden können.

Bei einem XML-Dokument kann im einfachsten Fall das Layout für die Onlineausgabe mit einem Cascading Stylesheet (CSS) definiert werden. CSS ist eine standardisierte Formatierungssprache für das Web, die ursprünglich für HTML definiert wurde. Ein CSS-fähiger Webbrowser, der auch XML-Quellen lesen kann, stellt das XML-Dokument am Bildschirm formatiert dar.

## Beispiel

Das getaggte Rezept für Donauwellen in Abbildung 1.6 kann mit Formatierungsangaben im Browser formatiert angezeigt werden:

Abbildung 1.7: Ausgabe des Rezepts für Donauwellen im Webbrowser

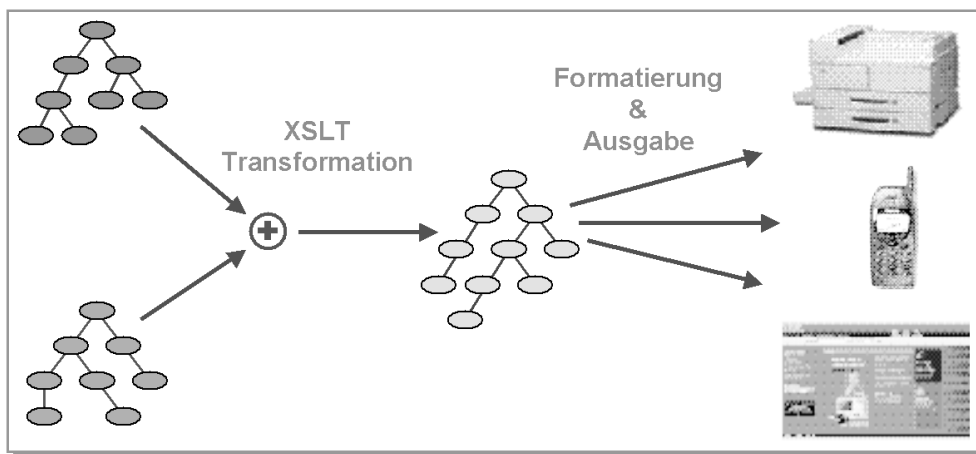


Abbildung 1.8: XML-Formatierung

Für anspruchsvollere Formatierungsaufgaben gibt es XSL (eXtensible Stylesheet Language). XSL enthält eine Transformationssprache und eine Formatierungssprache. Beide Sprachen sind XML-Sprachen.



Die Transformationssprache (XSLT) definiert Regeln, mit denen beschrieben werden kann, wie ein XML-Dokument in ein anderes transformiert wird. Damit sind beliebige Umformungen möglich. Das Zieldokument muss nichts mehr mit dem Ausgangsdokument gemeinsam haben.

Mit der Formatierungssprache (XSL-FO) wird die Formatierung beschrieben. Zugehörige generische Programme, XSLT-Prozessoren und Formatierer, sorgen für die Produktion der Ausgabedokumente.

### Beispiel

Aus dem getaggtten Rezept in Abbildung 1.6 kann auch eine ansprechende Druckausgabe erzeugt werden. Die Formatierungsangaben werden entsprechend ausgetauscht. Der getaggte Inhalt wird nicht verändert.

## Donauwellen

### Zutaten

*Boden*

250 g	Margarine
200 g	Zucker
375 g	Mehl
3	Eier
1 Pck.	Backpulver
1 Pck.	Vanillinzucker

*Belag*

1 l	Milch
2 Pck.	Puddingpulver
2 Pck.	Butterkekse
1/2 Pfund	Puderzucker
1/2 Pfund	Palmin
	Kakao
2 Gläser	Kirschen



### Zubereitung

1. Alle Zutaten für den Teig gut verrühren.
2. Auf ein gefettetes Backblech streichen und 2 Gläser Kirschen abtropfen lassen. Verteilen und bei 175 Grad 20-30 Minuten backen.
3. Von 1 Liter Milch mit 2 Päckchen Puddingpulver und 1 Ei. Zucker Pudding kochen, auf den gebackenen Boden streichen.
4. Butterkekse darauf verteilen und erkalten lassen.
5. 1/2 Pfund Puderzucker 2 Eier und 2 Esslöffel Kakao verrühren.
6. 1/2 Pfund Palmin erwärmen und tropfenweise unter die Masse geben.
7. Creme auf die Kekse verteilen.
8. Sehr fest werden lassen, am besten im Kühlschrank.

Abbildung 1.9: Druckausgabe des Rezepts für Donauwellen

Die Trennung von Inhalt und Darstellung ist eine wichtige Basis für Single Source Publishing:

- Die Formatierung kann weitgehend programmgesteuert erledigt werden.
- Derselbe Inhalt kann durch Austausch der Formatangaben unterschiedlich dargestellt werden.
- Dieselben Formatfestlegungen können konsistent für viele Inhalte verwendet und effizient geändert werden.

### 1.3.4 Formatierung ist nicht alles

Der Single-Source-Gedanke lässt sich über die Formatierung hinaus beliebig ausdehnen:

- Andere Bereiche wie Entwicklung, Marketing, Schulung können einbezogen werden und dieselben Quellen nutzen.
- Die eigentlichen Dokumentinhalte können durch Zusatzinformationen (Metadaten) angereichert werden, die für die Inhaltsverwaltung und -verwendung notwendig sind.
- Auf Basis der getaggtten Inhalte lassen sich vielfältige Nutzungsmöglichkeiten effizient realisieren.

### Beispiel

Für die Rezeptsprache lässt sich problemlos ein Umrechner programmieren, der bei Angabe der Personenzahl die Mengenangaben überall passend austauscht.

The image shows a recipe page for 'Donauwellen' with a calculator interface. The recipe is divided into three sections: 'Zutaten' (Ingredients), 'Boden' (Crust), and 'Belag' (Topping). The ingredients list includes 250g Margarine, 200g Zucker, 375g Mehl, 3 Eier, 1 Backpulver, 1 Vanillinzucker, 1l Milch, 2 Pck. Puddingpulver, 1 Pck. Butterkekse, 1/2 Pfund Puderzucker, Kakao, and 1/2 Pfund Palmin. The instructions are: 'Alle Zutaten für den Teig gut verrühren.', 'Auf ein gefettetes Backblech streichen und 2 Gläser Kirschen abtropfen lassen. Verteilen und bei 175 Grad 20-30 Minuten backen.', 'Von 1 Liter Milch mit 2 Päckchen Puddingpulver und 1 El. Zucker Pudding kochen, auf den gebackenen Boden streichen. Butterkekse darauf verteilen und erkalten lassen. 1/2 Pfund Puderzucker 2 Eier und 2 Esslöffel Kakao verrühren. 1/2 Pfund Palmin erwärmen und tropfenweise unter die Masse geben. Creme auf die Kekse verteilen.', and 'Sehr fest werden lassen, am besten im Kühlschrank.' The calculator interface includes icons for 'Zurück', 'Versenden', 'Ablagen', 'Drucken', 'Einkaufszettel', 'Bewerten', 'Nährwerte berechnen', and 'Frage stellen'.

Zutaten	
<i>Boden</i>	
250 g	Margarine
200 g	Zucker
375 g	Mehl
3	Eier
1	Backpulver
1	Vanillinzucker
<i>Belag</i>	
1 l	Milch
2 Pck.	Puddingpulver
1 Pck.	Butterkekse
1/2	Pfund Puderzucker
	Kakao
1/2	Pfund Palmin

**Donauwellen**

Zurück Versenden Ablagen Drucken Einkaufszettel Bewerten Nährwerte berechnen Frage stellen

Alle Zutaten für den Teig gut verrühren.

Auf ein gefettetes Backblech streichen und 2 Gläser Kirschen abtropfen lassen. Verteilen und bei 175 Grad 20-30 Minuten backen.

Von 1 Liter Milch mit 2 Päckchen Puddingpulver und 1 El. Zucker Pudding kochen, auf den gebackenen Boden streichen. Butterkekse darauf verteilen und erkalten lassen. 1/2 Pfund Puderzucker 2 Eier und 2 Esslöffel Kakao verrühren. 1/2 Pfund Palmin erwärmen und tropfenweise unter die Masse geben. Creme auf die Kekse verteilen.

Sehr fest werden lassen, am besten im Kühlschrank.

Abbildung 1.10: Rezeptaussgabe mit Umrechner

Ein weiteres Beispiel ist eine einfache mobile Anwendung, die aus den Rezeptzutaten für die Anzeige am Handy eine kompakte Einkaufsliste zusammenstellt, die beim Einkaufen abgerufen werden kann:

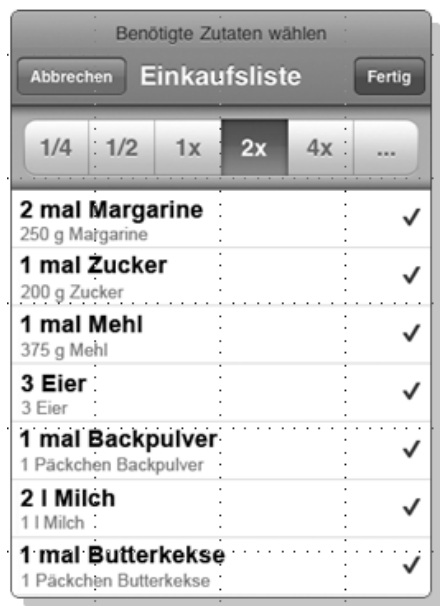


Abbildung 1.11: Anzeige der Rezeptzutaten am Handy

## 1.4 Die letzte Stufe: das Baukastenprinzip

Inhalte werden nicht nur in unterschiedlichen Darstellungen benötigt, sondern auch in unterschiedlichen Zusammenstellungen für unterschiedliche Zwecke und unterschiedliche Zielgruppen. Single Source Publishing heißt jetzt, dass einerseits gemeinsame Teile nur einmal erstellt und gepflegt werden, andererseits jedem Nutzenden nur die Informationen angeboten werden, die sie oder er benötigt, nicht mehr und nicht weniger.

Dieses Ziel ist mit einer starren Dokument- und Kapitelstruktur nicht zu erreichen. Für Single Source Publishing kommt dagegen eine Strukturierungsmethode zum Einsatz, die Inhalte in wiederverwendbare Module zerlegt, aus denen sich nach dem Baukastenprinzip nach Bedarf passende Dokumente zusammenstellen lassen.

Die Methode stammt aus der Hypertextforschung und wird in der Onlinehilfe-Welt seit vielen Jahren praktiziert. Für einen Baustein hat sich der Begriff „Topic“ etabliert. Technische Umsetzungen für Topics boten lange Zeit nur die Online-Autorenwerkzeuge.

Seit Mai 2005 gibt es die standardisierte Informationsarchitektur DITA (Darwin Information Typing Architecture). DITA bietet eine tool- und herstellerunabhängige Definition für Topics auf XML-Basis. Damit sind entscheidende Voraussetzungen für das Single Source Publishing erfüllt: die Trennung von Inhalt und Layout sowie eine geeignete Strukturierungstechnik.

Die Topic-orientierte Strukturierung ist die Basis für erfolgreiche Single-Source-Lösungen. Allerdings ist der Umgang mit dieser Strukturierungstechnik noch immer ungewohnt.

### 1.5 Irrungen und Wirrungen über Single Source Publishing

Nicht richtig ist	Sondern
Single Source Publishing heißt lediglich, unterschiedliche Ausgabeformate produzieren zu können.	Die Produktion unterschiedlicher Ausgabeformate ist nur ein Aspekt von Single Source Publishing. Ein weiterer zentraler Aspekt ist die möglichst flexible und weitreichende Verwendung der einmal erstellten Inhalte, sodass für jeden Bedarf genau die Informationen zur Verfügung gestellt werden können, die benötigt werden.
Single Source Publishing ist eine neue Erfindung.	Der Name ist noch recht jung, Single-Source-Lösungen gibt es aber schon, seit Informationen elektronisch erstellt, verwaltet und genutzt werden.
Für erfolgreiches Single Source Publishing reicht es aus, ein passendes Tool zu finden.	Erfolgreiches Single Source Publishing heißt primär, auf eine Topic-orientierte Strukturierung umzustellen und die monolithische Dokumentsicht aufzugeben. Die technische Umsetzung ist erst der zweite Schritt.

**Tabelle 1.2:** Irrungen und Wirrungen über Single Source Publishing

### 1.6 Das Wichtigste aus Kapitel 1

Single Source Publishing heißt:

- denselben Inhalt nur einmal erstellen
- vorhandene Inhalte so weit wie möglich wiederverwenden
- aus einem Inhaltsbestand Ausgabedokumente produzieren, die sich in Umfang, Zusammenstellung und Darstellung unterscheiden können
- denselben Inhalt in unterschiedlichen Layouts darstellen

Das Ziel, Inhalte an einer Stelle zusammenzuführen und sie dann mehrfach zu verwenden, wurde im Dokumentationsbereich in den letzten Jahren intensiv verfolgt, weil bei der redundanten Inhaltserstellung die Dokumentationsaufgaben nicht mehr zu bewältigen sind und die Kosten explodieren.

Publishing ist daher ein wichtiger Anwendungsbereich für Single-Source-Umgebungen, aber nicht der einzige. Auch in anderen Bereichen kann es sinnvoll sein, Inhalte zusammenzuführen, um Mehrfacharbeiten einzusparen und die Qualität zu verbessern.

Die Basis für Single-Source-Umgebungen ist eine Topic-orientierte Strukturierung.

Weitere wichtige Aspekte sind:

- Trennung von Inhalt und Layout
- Effiziente technische Unterstützung durch geeignete Tools

Der jüngste Trend, die mobile Dokumentation, macht Single Sourcing immer wichtiger. Das folgende Kapitel geht daher näher auf dieses Thema ein.

In den anschließenden Kapiteln dieses Buches geht es dann um die Topic-orientierte Strukturierung, die Basis für erfolgreiches Single Sourcing.



# 2

## Der jüngste Trend – Mobil

Die Erstellungsseite steht vor neuen Herausforderungen, denn Smartphone, iPad & Co haben die Dokuwelt wieder einmal kräftig aufgerüttelt. Was vor kurzem noch als jugendliche Spielerei belächelt wurde, entwickelt sich heute zur wichtigsten Kommunikationsfläche. Der ständige Austausch an jedem Ort ist den Nutzern sehr wichtig. Auch die Produktinformation wandert mehr und mehr in die kleinen Geräte, die wir immer mit uns tragen. Ohne lange Fehlerbeschreibungen kann unser Smartphone heute schon anhand eines seltsamen Geräuschs im Motor den Fehler identifizieren und uns direkt Abhilfe anbieten. Wir müssen keine dicken Bücher mitschleppen, nicht langwierig den Rechner hochfahren oder hilflos am Straßenrand stehen. Die Einsatzmöglichkeiten sind grenzenlos und viel versprechend.

Die mobilen Medien und ihre technischen Möglichkeiten entwickeln sich rasant. Sie lösen nicht unbedingt die bestehenden Medien ab, sondern müssen parallel neben den vorhandenen bedient werden. Die Folge dieser Entwicklung ist, dass Inhalte auf immer mehr Kanälen für unterschiedliche Zwecke, in unterschiedlicher Form und in immer kürzeren Zeiträumen bereitgestellt werden müssen.

Dieses Kapitel gibt einen Überblick über die Perspektiven, die sich durch die mobilen Medien der Dokuwelt eröffnen, und zeigt die Auswirkungen auf die Inhaltsauswahl, -gestaltung und -struktur. Auf technische Details wird nur im Überblick eingegangen.

### 2.1 Gerätevielfalt

Kennzeichnend für die mobilen Geräte sind zum einen geringe Größe und Gewicht, die es erlauben, das Gerät überall mitzunehmen, und zum anderen ein kleiner Bildschirm mit einer Auflösung von 240 x 320 bis derzeit maximal 640 x 940 (iPhone 4) und 1024 x 600 Pixel (Samsung Galaxy Tab).

Nach herkömmlichen Kriterien ist der Bildschirm eigentlich viel zu klein, um Inhalte sinnvoll darstellen zu können. Die kleine Anzeigefläche wird allerdings wettgemacht durch Zusatzfunktionen wie Wechsel von Hoch- und Querformat, komfortables Vergrößern und Verkleinern des Inhalts, automatisches Anpassen des Inhalts an die Anzeigefläche und immer ausgereifere technische Möglichkeiten, was Pixeldichte, Kontrastverhältnis und Material betrifft, sodass die Inhalte gestochen scharf erscheinen und gut lesbar sind.

Die Vielfalt der Geräte in der mobilen Welt stellt eine besondere Herausforderung für die Erstellungsseite dar. Ständig kommen neue Gerätetypen und Funktionen dazu. Special-

Purpose-Geräte, die für eine Funktion optimiert sind, wie das Mobiltelefon, entwickeln sich zu Multifunktionsgeräten, die vielseitige Funktionen anbieten. Was die Informationsdarbietung betrifft, unterscheiden wir Stand heute in Hinblick auf Größe, Verbreitungsgrad und Vielseitigkeit Smartphone, E-Book-Geräte und Tablet-PC.

### 2.1.1 Smartphone

Das kleine, handliche Smartphone ist mit Abstand das am weitesten verbreitete Mobilgerät und bietet die umfassendsten Vernetzungs- und Integrationsmöglichkeiten.

#### Geschichte

Vorläufer der heutigen mobilen Geräte waren die in den 1980er Jahren entwickelten Organizer mit Funktionen für Kalender-, Adress- und Aufgabenverwaltung. Anfang der 1990er Jahre wurde dieses Konzept in Form von PDAs (Personal Desktop Assistent) neben den Mobiltelefonen weiter entwickelt. Bekannt sind die Palm-Geräte des Herstellers Palm, Inc. mit dem Betriebssystem Palm OS.



Abbildung 2.1: Smartphone iPhone mit Apps



Die weitere technische Entwicklung führte dann zum Smartphone (intelligentes Telefon), das Mobiltelefon und PDA mit den klassischen Funktionen Telefonie, SMS, Adress- und Kontaktverwaltung in einem Gerät vereint. Erste prominente Vertreter waren die BlackBerry-Geräte des kanadischen Herstellers RIM, die sich vor allem an Geschäftsleute richten. BlackBerry gilt als Vorreiter des Push-Mail-Dienstes, der eingehende E-Mails ohne manuelles Abrufen direkt an ein mobiles Endgerät weiterreicht.

Apple hat 2007 mit seinem iPhone den Smartphone-Boom entfacht und gilt als Wegbereiter und Trendsetter. Der Erfolg beruht auf dem ansprechenden Design, der intuitiven Bedienung und der Möglichkeit, über Miniprogramme, so genannte Apps, das Gerät ganz individuell auszustatten. Apps können über passende Entwicklungsumgebungen (SDK – Software Development Kit) entwickelt und dann kostenfrei oder kostenpflichtig über spezielle App Stores zur Verfügung gestellt werden.

### **Betriebssysteme**

Mittlerweile gibt es eine Fülle von proprietären, kommerziellen und Open-Source-Betriebssystemplattformen für Smartphones.

Apple iOS ist ein proprietäres, geschlossenes Betriebssystem. Es ist ausschließlich auf Apple-Geräten zu finden, andere Gerätehersteller dürfen es nicht verwenden. Für die App-Entwicklung bietet Apple das iPhone SDK. Die Programmiersprache ist Objective C. Anwendungen werden im Apple App Store bereitgestellt, nachdem sie durch Apple geprüft wurden. Sollen Anwendungen kommerziell vertrieben werden, muss eine Lizenz käuflich bei Apple erworben werden.

Symbian ist eines der ältesten und am weitesten verbreiteten Betriebssysteme. Es läuft auf vielen Nokia-Smartphones sowie auf Geräten von Samsung und Sony Ericsson und bietet aktuell die wohl größte Produktvielfalt, von reinen Touchscreen-Modellen bis hin zu Geräten mit Schreibmaschinentastatur. Mit der neuesten Version wurde Symbian vom geschlossenen zum offenen System. Was nicht zum Standardrepertoire gehört, kann über den Nokia Ovi Store nachgerüstet werden.

2008 brachte Google das Android-Betriebssystem als offenes Betriebssystem auf den Markt, das besondere Stärken in Verbindung mit Google-Diensten, z. B. Google-Mail hat. Jeder Gerätehersteller darf es verwenden und weiterentwickeln. Viele Hersteller wie Acer, HTC, LG, Motorola und Samsung statten einen Teil ihrer Geräte damit aus. Es ist einfach zu bedienen und hat einen eigenen App Store, den Android Market. Die App-Entwicklung basiert auf der Programmiersprache Java.

Microsoft löste mit Windows Phone 7 sein mobiles Betriebssystem Windows Mobile ab. Mobile Varianten der wichtigsten Office-Anwendungen sind vorinstalliert und ermöglichen es, unterwegs mit Office-Dokumenten zu arbeiten. Smartphones mit Windows Phone 7 kommen von HTC, Samsung, LG und Dell. Für die App-Entwicklung gibt es eine kostenlose Version von Visual Studio 2010 Express für Windows Phones, mit speziellen Funktionen für mobile Anwendungen (Windows Phone Developer Tools RTW). Die Program-

miersprachen sind C# mit .NET sowie Silverlight und XNA für die Spieleentwicklung. Der Vertrieb der mobilen Anwendungen erfolgt über den Windows-Mobile-Marktplatz.

Samsung brachte 2010 Smartphones mit dem neu entwickelten, offenen Betriebssystem Bada auf den Markt, heute noch ohne eigenes Angebot für Zusatzprogramme. Es bietet neben den klassischen Funktionen wie Telefonie, SMS, Adress- und Kontaktverwaltung auch viele Funktionen für komfortables Webbrowsing. Außerdem unterstützt Bada Flash-Anwendungen, was es ebenfalls für den Einsatz im mobilen Web prädestiniert.

### Weiterentwicklung

Die neue Generation der Smartphones macht dem Computer zunehmend Konkurrenz. Mit Near Field Communication (NFC) wird das Smartphone zur universellen Steuerkonsole. Über das Gerät lässt sich die Autotür entriegeln, die Musikkwiedergabe regulieren, an der Supermarktkasse bezahlen. Es nimmt Fotos und Videos mit räumlicher Tiefe auf und dient gleichzeitig als Spielekonsole.

#### 2.1.2 E-Book-Lesegerät

Ein E-Book-Lesegerät ist ein portables Special-Purpose-Gerät für elektronische Bücher. Kindle von Amazon ist das bislang erfolgreichste E-Book-Lesegerät.



Abbildung 2.2: E-Book-Lesegeräte Kindle

Weitere bekannte Geräte sind Sony Reader PRS und Weltbild eBook-Reader. Ein E-Book-Lesegerät ist speziell ausgerichtet auf das Lesen digitaler Bücher und bietet dafür optimierte Bedingungen:

- Es ist leicht.
- Es ist preisgünstiger als Smartphone oder Tablet-PC.
- Es hat ein hoch auflösendes, flimmerfreies Display. Die dahinter liegende Technik ist unterschiedlich (E-Ink, Mirasol, LCD), aber in jedem Fall darauf ausgerichtet, möglichst stromsparend eine Darstellung zu erzielen, die auch bei Sonneneinstrahlung und wenig Licht gut lesbar ist. Die vom amerikanischen Unternehmen E-Ink entwickelte Technik „elektronische Tinte“ bietet eine kontrastreiche Darstellung von Texten und Grafiken in Schwarzweiß. Sie ähnelt in vielen Punkten der Druckerschwärze herkömmlicher Bücher und hat den Vorteil, nur beim Umblättern Energie zu verbrauchen.
- Es benötigt wenig Strom, sodass auch mehrere tausend Seiten ohne Nachladen des Akkus gelesen werden können.
- Es enthält passende Navigationsfunktionen wie Blättern im Buch und Suchen in der Bibliothek.

Natürlich geht auch bei diesen Geräten der Trend zum Mehrzweckgerät. Typische Erweiterungen bieten Anbindung ans Internet, Videoabspieelfunktion und Einbinden von sozialen Kommunikationsmitteln, um mit anderen Lesern Leseerfahrungen auszutauschen. Die in vielen Geräten integrierten Funkmodule ermöglichen von fast jedem Ort aus den Zugriff auf die Shops der Buchhändler. E-Book-Lesegeräte eignen sich auch als Ersatz für sperrige Tageszeitungen, da immer mehr Verlage ihre Publikationen auch digital in Formaten anbieten, die für E-Book-Lesegeräte tauglich sind. Modelle mit Touchscreen ermöglichen zum Teil auch handschriftliche Anmerkungen, die sich später extrahieren lassen.

### 2.1.3 Tablet-PC

Ein Tablet-PC ist ein tragbares Mehrzweckgerät, das viele Funktionen von Smartphone, E-Book-Lesegerät und klassischem Laptop vereint und noch weitere bietet. Ausgestattet mit einem größeren, gut lesbaren Display eignet sich das Gerät hervorragend zum Lesen, Filme anschauen und Surfen im Internet. Wie ein Smartphone lässt sich ein Tablet-PC mithilfe von Apps beliebig erweitern und individuell ausstatten.

Auch hier hat Apple 2010 mit seinem iPad den Trend gesetzt und bestimmt seither den Markt. Sony Tablets, Samsung Galaxy Tab, Weltbild Tablet PC, Lenovo ThinkPad Tablet, Toshiba Folio 100, Blackberry Playbook und Acer Iconia W500 sind weitere bekannte Geräte. Das Besondere am Tablet-PC ist seine Bedienbarkeit über Stift, Berührungsgesten oder Spracheingabe.