

Eine Diplomarbeit im Studiengang Audiovisuelle Medien an der  
Fachhochschule Stuttgart – Hochschule der Medien

## **MingCMS**

**SWF ohne Flash –  
Möglichkeiten zur Erstellung eines  
Flash Content Management Systems mit Ming**

Vorgelegt von Helmut Moritz am 16.06.2004 – Matrikelnummer 11341

Erster Prüfer:  
Zweiter Prüfer:

Prof. Dr. Johannes Schaugg  
Prof. Uwe Schulz

Ich erkläre, dass ich die vorliegende Arbeit –

**„MingCMS; SWF ohne Flash – Möglichkeiten zur Erstellung eines Flash Content Management Systems mit Ming“**

ohne fremde Hilfe verfasst habe. Alle verwendeten Quellen sind im Literaturverzeichnis aufgeführt.

Helmut Moritz,  
Stuttgart, den 15.06.2004

# MingCMS

## SWF ohne Flash – Möglichkeiten zur Erstellung eines Flash Content Management Systems mit Ming

1	Abstrakt	1
2	Einleitung, Zielgruppe	2
	2.1 Aufgabenstellung	2
	2.2 Zielgruppe	3
3	Technische und theoretische Grundlagen über Content Management Systeme, die zu verwendenden Technologien und die Benutzung von Ming- Elementen	5
	3.1 Was ein CMS leisten könnte und was MingCMS leisten soll	5
	3.1.1 Anzahl und Struktur der Inhaltseiten	5
	3.1.2 Bedarf an regelmäßigen Inhaltsaktualisierungen	6
	3.1.3 Wunsch nach Trennung zwischen Inhalt und Layout	7
	3.1.4 Geringe Internetkenntnisse der Autoren	7
	3.1.5 Anzahl der Betreuer einer Seite und deren Organisation	8
	3.1.6 Vergabe von Zugriffsrechten	9
	3.1.7 Lebenszyklus der Inhalte	10
	3.1.8 Wiederherstellung alter Zustände der Website	12
	3.1.9 Art der Inhalte	12
	3.1.10 Anbindung externer und interner Datenquellen	14
	3.1.11 Zwei Versionen – HTML und Flash	15
	3.1.12 Flash innerhalb von HTML integriert	15
	3.1.13 Anzeige auf verschiedenen Ausgabegeräten	16
	3.1.14 Barrierefreiheit	16
	3.2 Bestehende Systeme und Anwendungen	17
	3.3 Macromedia Flash	20
	3.3.1 FlashCMS-Möglichkeiten mit Macromedia-Technologien	20
	3.4 Das SWF-Format	21
	3.5 ActionScript	22
	3.6 PHP	24
	3.7 Ming	25
	3.7.1 Ming unter PHP	25
	3.7.2 Ming unter anderen Sprachen	26
	3.8 Ming-Elemente	27
	3.8.1 Erzeugung eines SWF-Movies	27
	3.8.2 Darstellung der Einzelelemente	28
	3.8.2.1 Text	28
	3.8.2.2 Grafik	29

	3.8.2.3 Bitmap	30
	3.8.2.4 Schaltfläche	31
	3.8.2.5 Animation	32
	3.8.2.6 MovieClips – Sprites	32
	3.8.2.7 ActionScript	33
	3.8.2.8 Erzeugung einer automatisierten Navigation	33
3.9	OpenSource	36
	3.9.1 Voraussetzungen	36
	3.9.2 Lösungsmöglichkeit	37
4	Konzept für ein CMS als „echtes Autorensystem“ für die definierte Zielgruppe unter Verwendung der beschriebenen Technologien	38
	4.1 Funktionsweise des MingCMS für den Benutzer	38
	4.2 Benutzeroberfläche	39
	4.3 Klassische Teilung oder alles in einer Hand?	43
	4.3.1 Notwendigkeit zur Auflösung etablierter Strukturen für MingCMS	43
5	Konkrete Realisierung des MingCMS	46
	5.1 Wie das Content Management System prinzipiell arbeiten sollte	46
	5.1.1 Architektur von MingCMS	47
	5.1.2 Datenorganisation	49
	5.1.3 Verarbeitung	52
	5.1.3.1 Erstellung der Seite	52
	5.1.3.2 Generierung bei Anfrage	53
	5.1.4 Ausgabe/Benutzeroberfläche	54
	5.2 Notwendigkeit zur Dokumentation	54
	5.3 Ergänzende Technologien	55
	5.3.1 Apache-Webserver	55
	5.3.2 XML	56
	5.3.3 MySQL-Datenbanken	57
6	Aufwandsschätzung, Zusammenfassung und Ausblick auf weitere Verwendungsmöglichkeiten von Ming	58
	6.1 Realisierbarkeit des MingCMS	58
	6.1.1 Aufwandsschätzung	59
	6.2 Grenzen in der Erweiterbarkeit	60
	6.3 Fähigkeiten und Grenzen von Ming	61
	6.3.1 Handhabung und Organisation von Elementen und Daten	61
	6.3.2 Animationsmöglichkeiten	62
	6.4 Zusätzliche Anwendungsmöglichkeiten für Ming	63

## Literaturverzeichnis

## Anhang

## 1 Abstrakt

Möglichkeiten zur Erstellung eines Flash-Content Management Systems mit Ming – ausgehend von dem Untertitel der Arbeit beschäftigt sich dieses Dokument zunächst mit der Aufgabenstellung und definiert die Zielgruppe. Dem schließt sich eine Betrachtung über Content Management Systemen (CMS) an, da vor der Realisierung eines solch komplexen Systems ergründet werden muss, welche Komponenten zu einem CMS gehören und was eine solche Software prinzipiell ausmacht. Überlegungen zur Gestaltung einer Benutzeroberfläche, der Schnittstelle zwischen Benutzer und CMS, runden die ersten Kapitel ab. Weiter soll exemplarisch dokumentiert werden, welche Systeme aktuell auf dem Markt angeboten werden, die einem Flash-CMS zumindest teilweise entsprechen. Eine Beschäftigung mit den vorhandenen Produkten von Macromedia soll zeigen, welche Lösungen derzeit bei der Erstellung von dynamischen und animierten Websites aktuell sind. In einem technischeren Abschnitt stehen das swf-Format, die flashinterne Programmiersprache ActionScript, die Skriptsprache PHP und anschließend deren Ming-Klassen im Mittelpunkt. Außerdem soll der Frage nachgegangen werden, ob sich ein Flash-CMS, mit dem der Benutzer die Möglichkeit erhalten soll, die bisher von Agenturen bewältigten Aufgaben selbst zu übernehmen – also ein „echtes“ Autorensystem“, überhaupt sinnvoll realisierbar ist.

Auf welche Weise Ming arbeitet, welche Sprachelemente es beinhaltet und wie einzelne Elemente erzeugt und behandelt werden können, wird anschließend anhand von konkretem Quellcode dokumentiert. Der stärker praktisch orientierte Teil beschäftigt sich mit der Realisierung eines Flash-CMS mit Ming und zeigt mögliche Systemarchitekturen. Eine Zusammenfassung beleuchtet die Realisierbarkeit und die technischen Grenzen eines Flash-CMS in der angedachten Form und weist auf zusätzliche Anwendungsmöglichkeiten von Ming hin.

## 2 Einleitung, Zielgruppe

Die Entwicklung von Systemen, die Inhalte für Internetanwendungen aufbereiten und organisieren können, ist die Hauptaufgabe für Webentwickler. Die Intensität des Aufwandes variiert je nach Anspruch an Zusammensetzung und Aussehen, dem Umfang des aufzubereitenden Inhaltes und den Fertigkeiten desjenigen, der die Inhalte nach der Fertigstellung des Systems produziert, sammelt und für die Anwendung aufbereitet.

Eine große Hilfe zur Organisation und Aufbereitung von Daten für Internetanwendungen sind Content Management Systeme (CMS). Allein auf dem deutschen Markt existieren derzeit über 1600 solcher Systeme, die sich in zahlreichen Eigenschaften und Fähigkeiten voneinander unterscheiden <cont1>.

Content Management Systeme zur Aufbereitung von Flashseiten sind im Gegensatz zu der großen Anzahl existierender Produkte sehr selten. Es existieren nur einige wenige Systeme. Deren Verwendung ist gewöhnlich mit Kosten für Lizenzen verbunden. Und die Installation und Einrichtung kann eigentlich nur von einer Agentur oder dem Anbieter selbst übernommen werden.

Diese Arbeit möchte der Frage nachgehen, ob es möglich und sinnvoll ist, ein frei verfügbares CMS zu entwickeln, mit dem ein unbedarfter Anwender ohne Agenturunterstützung eine dynamische und animierte Flashanwendung für das Internet gestalten und entwickeln kann. Dabei steht die Technik Ming in Kombination mit der Skriptsprache PHP neben allgemeinen Überlegungen zu Content Management Systemen im Mittelpunkt.

### 2.1 Aufgabenstellung

Die Möglichkeiten zur Realisierung eines Content Management Systems, mit dem animierte Flash-Seiten erstellt werden können – das ist die Grundidee dieser Arbeit. Seit der Version 4.05 der Skriptsprache PHP ist eine Erweiterung dem Namen Ming integriert worden, bei der es sich um eine Bibliothek zur Erstellung von SWF-Flash-Movies handelt. „Ming is an open-source [...] library which allows you to create SWF [...] format movies“ <php-h1>. Da bisher keine komplexen Internetseiten oder Anwendungen mit Ming im Web zu finden sind, gilt es zunächst zu klären, ob es mit dieser Technik

möglich ist eine umfangreiche Internetseite mit einer automatisierten Navigation zu erstellen. Aus diesem Grund wird diese Diplomarbeit durch einen praktischen Teil ergänzt, der die technischen Möglichkeiten von Ming, orientierend an der Fragestellung, erforscht.

Zu den weiteren Aufgaben gehören eine Diskussion über die Anforderungen an ein CMS, basierend auf den Überlegungen zur Zielgruppe und zu den entsprechenden Anwendungen sowie die Ermittlung der theoretischen und technischen Anforderungen. Und schließlich sollte noch eine Diskussion darüber geführt werden, welche Grenzen Ming besitzt und welche weiteren Anwendungsmöglichkeiten mit dieser Technik denkbar sind.

## 2.2 Zielgruppe

„Es gibt nicht das einzig richtige WCMS [Web Content Management System], sondern viele Systeme, die alle für ihren jeweiligen Bereich spezialisiert sind.“ <krue 2002, S.38>  
Und tatsächlich präsentiert sich der CMS-Markt äußerst variabel und die Produkte unterscheiden sich in zahlreichen Punkten wie Technologie, Lizenzmodell, Zielgruppe, Erweiterbarkeit, Datenimport, Personalisierung, Workflow, Mehrsprachigkeit, Automatisierungsmöglichkeiten und vielem mehr.

Allein eine Unterteilung der unterschiedlichen Ausrichtungen von Content Management Systemen bedeutet eine eigene Herausforderung für sich. Das Online-Informationportal [contentmanager.de](http://contentmanager.de) unterscheidet die Systeme bspw. nach:

Enterprise Content Management  
Open-Source-Lösungen  
Agenturlösungen  
Groupware-Lösungen  
Redaktionssysteme  
ASP-Lösungen  
Intranet-Lösungen

*Einteilung unterschiedlicher Content Management Systeme*

- Enterprise Content Management (Anspruchsvolle Highend-Lösungen mit starker Integration in Back-End-Systeme),
- Open-Source-Lösungen (Lösungen basierend auf der Open-Source-Lizensierung mit offengelegtem Quellcode);
- Agenturlösungen (Lösungen von Multimedia- und Web-Agenturen für den Projekteinsatz),
- Groupware-Lösungen (Lösungen, die auf die bestehende Infrastruktur von Groupware, wie z.B. Notes, aufsetzen),

- Redaktionssysteme (Umfangreiche Lösungen für redaktionelle Websites und Firmenpräsenzen),
- ASP-Lösungen (Lösungen auf Mietbasis, die gegen Gebühr bei einem Provider gehostet werden) und
- Intranet-Lösungen (Lösungen, die speziell auf die Anforderungen von Intranets vorkonfektioniert sind). <cont2>

Je nach Ausrichtung der Systeme orientiert sich das jeweilige CMS auch an einer unterschiedlichen Benutzergruppe (Großunternehmen, mittelständische Betriebe, Kleinunternehmen, Redaktionen, etc.) und wird von Benutzern mit variierenden Kenntnissen betrieben. Allerdings richten sich konventionelle CMS gewöhnlich nicht an einzelne Personen ohne einen internettechnologischen Hintergrund. Dies liegt daran, dass ein CMS in den meisten Fällen dazu konzipiert ist, größere Datenmengen zu handhaben, die üblicherweise in größeren Unternehmen oder beim Betrieb von umfangreichen Websites anfallen.

Die Idee, die dem in dieser Arbeit angestrebten System zugrunde liegt, fällt in die Gruppe der Open Source Lösungen (da der Quellcode offen sein soll und das System zur Weiterentwicklung gedacht ist) und im weiteren Sinne in die Gruppe der Agenturlösungen mit umgewandelter Rollenverteilung. Denn hier soll die Aufgabenteilung äquivalent zu Agenturlösungen sein, jedoch liegen beide Aufgaben letztendlich in einer Hand, können aber auch zwischen einem Produzenten und einem Pfleger der Seite aufgeteilt werden.

Das MingCMS, so der hier verwendete Arbeitsname, soll sich an Einzelbenutzer richten, die aus einem beliebigen Grund eine Internetpräsenz anstreben und hierbei einen animierten Auftritt wünschen. Diese kann bspw. zur Präsentation eines eigenen Produktes (z.B. <http://www.meeresrausch.com>), einem Unternehmensauftritt (z.B. <http://www.borkum-zweirad.de>), einem Künstlerportrait (z.B. <http://www.nosports.de>), einer Vereinsseite (z.B. <http://www.djk-singen.de>) oder einer persönlichen Vorstellung (z.B. <http://www.klaus-haupt.de>) dienen. (Datum des letzten Besuches der jeweiligen URL: 01.03.2004). Der Benutzer soll keine Kenntnisse über die Programmierung und Entwicklung einer Internetseite besitzen müssen, um mit MingCMS eine Website aufbauen zu können. Zudem soll es einem unbewanderten Benutzer möglich sein, das System auf einem konventionellen Rechner zu installieren und zu betreiben.



### 3 Technische und theoretische Grundlagen von Content Management Systemen, die zu verwendenden Technologien und die Benutzung von Ming-Elementen

#### 3.1 Was ein CMS leisten könnte und was MingCMS leisten soll

Anzahl und Struktur der Inhaltsseiten  
 Bedarf an regelmäßigen Inhaltsaktualisierungen  
 Wunsch nach Trennung zwischen Inhalt und Layout  
 Geringe Internetkenntnisse der Autoren  
 Anzahl der Betreuer einer Seite und deren Organisation  
 Vergabe von Zugriffsrechten  
 Lebenszyklus der Inhalte  
 Wiederherstellung alter Zustände der Website  
 Art der Inhalte  
 Anbindung externer und interner Datenquellen  
 Zwei Versionen: Flash und HTML  
 Flash innerhalb von HTML integriert  
 Anzeige auf unterschiedlichen Ausgabegeräten  
 Barrierefreiheit

Webanwendungen stellen höchst unterschiedliche Anforderungen an ein Content Management System. Jede Anwendung benötigt dabei ein CMS, das in der Lage ist, die Internetseite mit allen Bedürfnissen erstellen und verwalten zu können. Da zahlreiche Kriterien und Anforderungen für Websites existieren, muss zunächst eine Bestimmung erfolgen, welche Bedürfnisse bestehen. Soll die Seite z.B. auf unterschiedlichen Ausgabegeräten betrieben werden? Spielt Barrierefreiheit eine Rolle? Wer pflegt die Seite? usw.

##### *Anforderungen an ein Content Management System*

Allgemein gilt der Wunsch nach einer Trennung zwischen Inhalt und Layout als

Hauptgrund für die Verwendung eines CMS. Weitere Kriterien sind z.B. die Anzahl der Inhaltsseiten und der Bedarf an regelmäßigen Updates. Je umfangreicher, bzw. wichtiger diese Parameter werden, desto eher lohnt es sich, ein CMS einzusetzen. Zu diesen Parametern addieren sich weitere Kriterien wie die Anzahl der Betreuer einer Seite sowie deren Organisation, der Lebenszyklus der Inhalte, die Art der Inhalte, eine Vergabe von Zugriffsrechten und eine Anzeige auf verschiedenen Ausgabegeräten. Allgemein soll hier eine nähere Betrachtung auf diese Kriterien geworfen werden. Dazu wird diskutiert, welche Fähigkeiten das MingCMS aufweisen sollte. Diese ergeben sich aus der Zielgruppe und den angestrebten Einsatzgebieten (siehe 2.1 und 2.2).

##### 3.1.1 Anzahl und Struktur der Inhaltsseiten

Verfügt eine zu realisierende Website über viele Inhaltsseiten, ergibt sich relativ schnell der Bedarf nach einem CMS. Als Richtwert könnte die Zahl von 100 Inhaltsseiten und mehr gelten <zogg, 2003>. Neben der Anzahl der Seiten ist auch deren Struktur von Bedeutung. Sollen z.B. viele Inhaltsseiten untereinander verlinkt oder eine Suchfunktion

integriert werden, steigt ebenfalls die Notwendigkeit zur Verwendung eines CMS. „Die Wartung, Indexierung, Verschlagwortung, Verlinkung etc. von sehr großen Informationsbeständen wird durch ein CMS erheblich erleichtert, da fast alle Aufgaben automatisiert werden können.“ <krue 2002, S.29>

MingCMS:

- Fähigkeit zur Verwaltung einer großen Anzahl von Inhaltsseiten, wie z.B. zehn Inhaltsbereiche mit je zehn Unterseiten gleich hundert Seiten.
- Funktion zur Verlinkung.
- Funktion zur Verschlagwortung (Indexierung) und damit Integration einer Suchfunktion.

### 3.1.2 Bedarf an regelmäßigen Inhaltsaktualisierungen

Als Richtwert für die Aktualisierungshäufigkeit kann der Zeitraum von jeweils einer Woche angesehen werden - also mindestens einmal pro Woche der Inhalt des Webauftritts auf den neusten Stand gebracht wird <zogg, 2003>. News- und Börsendienste erreichen sicherlich die kürzesten Intervalle an Aktualisierungen, eine Seite mit den Zeiten von Kinovorführungen würde z.B. mindestens genau diese eine Woche erfordern. Dagegen müsste z.B. eine Liste mit Unternehmensfilialen seltener aktualisiert werden (siehe 3.1.7).

Wird eine neue Inhaltsseite eingepflegt, sollten sich bei manchen Anwendungen zusätzliche Teilbereiche, die auf die neue Seite hinweisen (z.B. Teaser – ein verlinkter Bereich auf einer Seite, der eine kurze Vorschau über den Inhalt hinter dem Link gibt), entweder automatisch anpassen oder einfach manuell anpassen lassen. Neben solch einer Aktualisierung kann ein CMS den Vorteil bieten, Inhalte zu festgelegten oder zufälligen Terminen automatisch zu variieren. Beispiele hierfür wären die unterschiedliche Aufbereitung der Inhalte nach Tages- oder Jahreszeit zur emotionalen Ansprache des Seitenbesuchers oder auch eine zufällige Inhaltssteuerung zur Spannungssteigerung beim Besucher.

Werden Inhalte regelmäßig aktualisiert, besitzt eine Seite einen stetigen Zustrom an Daten. Auf der anderen Seite werden in einem stetigen Turnus nicht mehr aktuelle Inhalte aus der Seite heraus genommen. Hierbei ist zu klären, ob die Daten archiviert werden sollen oder nicht (siehe 3.1.7).

MingCMS:

- Die einfache Einpflege neuer Inhalte ist eine Grundvoraussetzung
- Von anderen Seiten soll automatisch auf die neue Seite verlinkt werden können
- Automatische Steuerungsfunktionen, welche die Tageszeit berücksichtigen oder zufällig Inhalte streuen sollen integriert werden können.
- Entfernte Inhalte sollen archiviert werden können.

### 3.1.3 Wunsch nach Trennung zwischen Inhalt und Layout

Der Einsatz eines CMS sollte letztlich immer dazu führen, dass der Inhalt einer Seite komplett vom Layout entkoppelt wird. Dadurch wird für Autoren im Idealfall aus dem komplexen Vorgang der Erstellung einer Website, was bei nur leicht angehobenen Ansprüchen an Aussehen und Funktionalität umfangreiche Entwicklungskennntnisse erfordert, ein übersichtlicher und einfacher Prozess. Wurde ein Layout einmal erstellt und technisch realisiert, kann ein Autor beliebig Inhalte in die Seite einpflegen. Darüber hinaus können die Bestandteile einer Website so zerlegt werden, dass sich Techniker, Grafiker und Autoren nur um ihre jeweiligen Kernkompetenzen kümmern müssen <zsch 2002, S. 61>.

Innerhalb eines schlichten CMS, an dem möglicherweise nur eine Person arbeitet, stellt sich eine andere Aufgabe: Welches Wissen sollte der Benutzer besitzen, um möglichst einfach mit dem System arbeiten zu können? Hier könnte eine Anlehnung in so genannten WYSIWYG-Editoren („What you see is what you get“) für Internetseiten gefunden werden (Dreamweaver, NetObjects Fusion, Frontpage). In diesen Entwicklungsumgebungen wird gewöhnlich. zunächst ein Layout entwickelt, in das später die jeweiligen Inhalte eingebettet werden.

MingCMS:

- Realisierung einer Entkopplung zwischen Layout und Inhalt.
- Ein Benutzer entwirft das Layout und pflegt Inhalte selbst ein.

### 3.1.4 Geringe Internetkenntnisse der Autoren

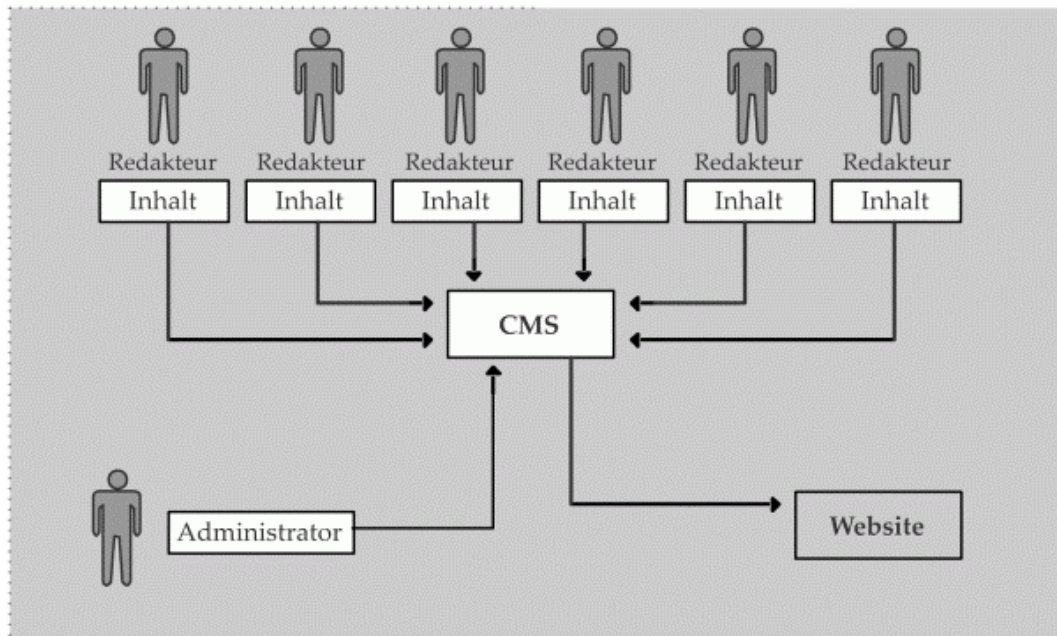
Innerhalb komplexer Webanwendungen (z.B. das Intranet eines größeren Unternehmens) pflegen zahlreiche Autoren Inhalte ein, die keine oder nur geringe Kenntnisse über das Internet besitzen. Für diesen Personenkreis sollte die Eingabe auf eine möglichst einfache Weise, z.B. über simple Formulare mit Feldern für Überschrift, Unterüberschrift, Einleitung, Text, Grafik oder Verlinkung erfolgen. Alle Formatierungen, die für die

Einpassung in das Layout erforderlich sind, werden über ein CMS realisiert. Für eine einfache Anwendung stellt sich hier die identische Frage wie im Punkt zuvor. Die sinnvolle Anwendung steht oder fällt mit den Fähigkeiten des Benutzers. Deshalb sollte die Schnittstelle des Anwenders für die Eingabe von Daten (Content und Formatierungsangaben zum Layout) möglichst einfach gehalten werden <krue 2002, S. 32>.

MingCMS:

- Einfache und klar definierte Formulare zur Eingabe von Daten für den Inhalt.

### 3.1.5 Anzahl der Betreuer einer Seite und deren Organisation



*Schema des Arbeitsablaufes für die Veröffentlichung einer Website mit einem CMS <krue>*

Je komplexer eine Website ausfällt und je mehr Personen sich um diese Seite kümmern, desto höher sind die Anforderungen an das verwendete CMS. Klassisch sieht die Organisation so aus, dass mehrere Autoren unterschiedliche Inhalte produzieren und ein Administrator sich um die Funktionalität der technischen Ressourcen kümmert (siehe Grafik). Durch solch ein Schema entstehen jedoch spezielle Probleme. Soll z.B. das CMS jeden neuen Inhalt sofort veröffentlichen? Dürfen zwei Autoren gleichzeitig an einer bestehenden Seite arbeiten? Für solche organisatorischen Probleme sollte das CMS klare Benutzerrechte und Strukturen besitzen. Ist z.B. ein Dokument gerade zur Bearbeitung geöffnet, bekommen „weitere Autoren, die gleichzeitig schreibend auf das Dokument zugreifen wollen, [...] einen Hinweis, dass der Zugriff momentan nur lesend stattfinden

kann.“ <krue 2002, S. 30> Eventuell kann es sinnvoll sein, in der Systemhierarchie die Rolle eines Redakteurs über die der Autoren zu stellen. Dieser überprüft die von den Autoren erstellten Inhalte und schaltet sie schließlich zur Veröffentlichung frei.

Für kleinere Websites mit nur einem Benutzer entfällt eine solche Unterscheidung von Rollen; und die Gefahr von mehreren gleichzeitigen Überarbeitungen an einem Dokument ist nicht vorhanden, da nur jeweils eine Person an der Website arbeiten kann.

MingCMS:

- Es ist keine Rollenverteilung erforderlich, da nur ein Benutzer kann an der Seite arbeiten kann. Dies ist für das angestrebte System vollkommen ausreichend, da nur ein relativ kleiner Content verwaltet werden soll und es sich an einzelne Benutzer richtet.

### 3.1.6 Vergabe von Zugriffsrechten

Unter den zahlreichen Anwendungsmöglichkeiten eines CMS sollen hier exemplarisch zwei beschrieben werden, die einen möglichen Bedarf nach Zugriffsrechten verdeutlichen sollen.

(1) Bei der aktuellen Pflege einer Nachrichtenseite (z.B. [www.spiegel.de](http://www.spiegel.de), [www.faz.net](http://www.faz.net)) stellt sich das gleiche Problem wie in einer klassischen Zeitungsredaktion: Welche der zahlreichen Nachrichten wird an welcher Stelle und zu welchem Zeitpunkt positioniert? Die Nachrichten werden dabei von Journalisten geschrieben und illustriert. Die Entscheidung über die Platzierung fällt z.B. ein Redakteur. Im Umkehrschluss sollte also der ‚einfache‘ Journalist erst gar nicht die Möglichkeit besitzen, Artikel freischalten zu können und dadurch öffentlich zu platzieren. Ein ausgereiftes CMS wird folglich dem Journalisten eine andere Benutzeroberfläche anbieten als dem Redakteur. Verschiedene Rollen wie Administration, Portal Manager, Content Manager und Redakteur können in jeweilige Benutzergruppe zusammengefasst und mit entsprechenden Rechten versehen werden.“ <ziet>

(2) Innerhalb eines komplexen Shopsystems (z.B. [www.amazon.de](http://www.amazon.de), [www.bol.de](http://www.bol.de)) könnten die Mitarbeiter davon profitieren, wenn ihnen entsprechend ihrer Rolle in der Unternehmensstruktur eine individuelle Oberfläche angeboten wird. Kümmt sich jemand um die Erstellung von Hintergrundinformationen zu Produkten, benötigt er einen anderen Datensatz und eine andere Editieroberfläche als die Person, die sich um Rabattaktionen oder Sonderangebote kümmert. Auch ein Einkäufer wird andere Daten

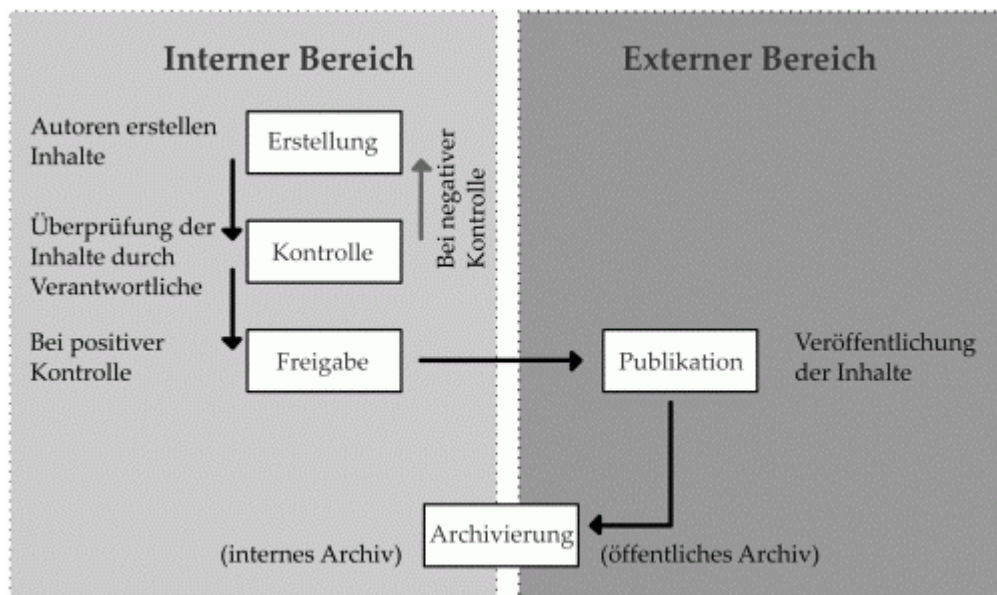
und Formulare benötigen, ebenso die Geschäftsleitung, die „eigentlich viel lieber sehen [möchte], wie viel denn schon verkauft wurde.“ <krue 2002, S. 31>

Diese Szenarien verdeutlichen, dass der eigentliche Sinn eines Zugriffsrechtes weniger die Vorenthaltung von Möglichkeiten ist, sondern viel mehr die Konzentration von Inhalten und Möglichkeiten auf die unterschiedlichen Rollen und Positionen innerhalb komplexer Strukturen und Arbeitsabläufe. Verwaltet ein CMS die Daten nach einer logischen Hierarchie und Ordnung, spielt bei den Zugriffsrechten und Rollenverteilungen die menschliche Logik die entscheidende Rolle. Ein CMS ist nicht in der Lage zu erkennen, ob die Nachricht über einen Regierungswechsel in Deutschland wichtiger ist als die Veröffentlichung eines neuen Buches von Boris Becker. Die im CMS vergebene Rolle des Redakteurs erkennt als Person die Bedeutungen und setzt die entsprechenden Prioritäten, wodurch wichtige Nachrichten oder Sonderangebote an einer extrovertierteren Stelle platziert werden.

MingCMS:

- Die oben diskutierten Punkte entstehen durch die Verwaltung größerer Datenmengen durch mehrere Personen. Für MingCMS sind keine Zugriffsrechte erforderlich, da das System nur von einer Person verwaltet werden soll und der Content überschaubar bleiben soll.

### 3.1.7 Lebenszyklus der Inhalte



*Existenzzyklus von Webinhalten* <zsch 2002, S. 56>

Jeder Inhalt einer Internetseite durchläuft von der Idee bis zur Löschung verschiedene Stationen. „Das Konzept des Content Life Cycle beschreibt alle diese Stationen, die z.B. ein bestimmter Text durchläuft, bevor er im Web zu sehen ist.“ <zsch 2002, S. 54ff> Nach diesem Konzept beginnt der Lebenszyklus mit dem Abschnitt der Erstellung, in dem die Idee durch den Autor mit einem entsprechenden Werkzeug, wie z.B. einem Texteditor oder einer Grafikanwendung, realisiert wird. Der Erstellung folgt eine Kontrolle des Inhaltes, die durch eine dem Autor übergeordnete Instanz durchgeführt wird und eine wichtige Station für die Qualität der auf der Website befindlichen Inhalte bedeutet. Kommt die Kontrolle zu einer positiven Beurteilung, wird der neue Inhalt genehmigt (Freigabe); fällt das Urteil negativ aus, wird das Dokument an den Autor zurückgegeben (Wiedervorlage). Ist der Inhalt schließlich autorisiert, gelangt er zur Publikation. Dabei wird der neue oder bearbeitete Inhalt publiziert und kann auf der Website öffentlich eingesehen werden. Auf seiner letzten Station landet der Inhalt im Archiv. „Die Archivierung alter Inhalte [...] hat einige bedeutende Vorteile: Man kann alte Stände der Website einsehen oder wiederherstellen oder auf Basis alter Beiträge neue der gleichen Art erstellen, indem man die alten als Vorlage nutzt.“ <ebd.>

Aus dieser Beschreibung eines Inhaltszyklus können verschiedenen Aufgaben und Tätigkeiten innerhalb eines CMS abgeleitet werden (siehe Tabelle). Diese Auflistung wird ergänzt durch nützliche Zusatzfunktionen. So existieren zur Erstellung eines Dokumentes verschiedene Varianten: komplette Neuerstellung sowie Import oder Ausleihe eines externen Dokuments. Innerhalb der Erstellung und der Kontrolle sollte eine Vorschau auf das Dokument möglich sein, um den Inhalt eingebunden in die fertige Seite beurteilen zu können. Ist das Dokument publiziert, sollte es auch gelöscht werden können und schließlich kann eine Möglichkeit zur Recherche im aktuellen Bestand oder im Archiv die Arbeit an den Inhalten optimieren <ziet>.

Dokument neu erstellen  
Content externer Anbieter importieren sowie Exportmöglichkeiten  
Dokument ausleihen  
Bearbeiten des Dokuments  
Zurückgeben  
Vorschau auf das Dokument  
Redigieren und Freigeben des Dokuments  
Publizieren des Dokuments (Übergabe des Dokuments von der Produktionsdatenbank an die Publikationsdatenbank, Spiegelung)  
Löschen des publizierten Dokuments und damit Archivierung  
Löschen des publizierten Dokuments und damit Archivierung  
Recherchieren in Bestand und/oder Archiv

**Tätigkeiten innerhalb eines CMS**  
<zsch 2002, S. 54ff>

MingCMS:

- Es werden keine Elemente zur Kontrolle (Freigabe, Wiedervorlage) benötigt, da der Lebenszyklus der Dokumente hier mit der Erstellung einschließlich einer Vorschaufunktion beginnt, dem sich unmittelbar die Veröffentlichung anschließt.
- Dem Betreuer soll es ermöglicht werden, die Dokumente sofort oder zu einem terminierten Zeitpunkt zu veröffentlichen.
- Wird ein Inhalt gelöscht soll dieser parallel in einem Archiv abgelegt werden.
- Eine Recherchemöglichkeit in alten Inhalten soll ebenso ermöglicht werden wie die Wiederherstellung von alten Seiteninhalten.

Die beiden zuletzt genannten Elemente sind sicherlich nicht zwingend notwendig, um ein kleines CMS sinnvoll zu verwenden. Jedoch tragen sie zu einer Vereinfachung des Arbeitsablaufes bei und rechtfertigen die Verwendung des Begriffes CMS im Zusammenhang mit der Software.

### 3.1.8 Wiederherstellung alter Zustände der Website

Neben den bereits erläuterten Vorteilen einer Archivierung alter Zustände der Website entstehen durch eine solche Funktion weitere Vorzüge. Wurde z.B. ein Update durchgeführt, mit dem der oder die Benutzer nicht zufrieden sind, kann bis zur erneuten Überarbeitung die alte Version angezeigt werden <krue 2002, S. 33>. Dadurch werden experimentelle Updates erleichtert, bei denen z.B. ergründet werden soll, wie ein veränderter Bereich der Seite im Nutzerverhalten Anklang findet. Der Weg zur Ausgangssituation bleibt jederzeit erhalten, da das CMS eine alte Version wiederherstellen kann. Und nicht unbeachtet bleiben soll der Nostalgiefaktor. Die Würdigung der Historie eines etablierten Webauftritts gestaltet sich nachhaltiger, wenn alte Versionen der Seite aufbereitet und neu präsentiert werden können.

MingCMS:

- Um ein Update experimentell ausprobieren zu können oder ein Update überarbeiten zu können, soll eine Wiederherstellung alter Zustände möglich sein. Hierbei solltet es ausreichend sein, das jeweiligen SWF-Movie mit den Datensätzen und externen Medien mit einer Datumsangabe zu speichern und in ein Archiv zu legen.

### 3.1.9 Art der Inhalte

Der Content einer Website kann durch eine Klassifizierung von Informationstypen und Änderungsfrequenzen unterteilt werden. „Grundsätzlich kann man sämtliche [Inhalte]



drei Informationstypen zuordnen: statische Informationen, dynamische Informationen und semidynamische Informationen“ <zsch 2002, S. 42ff>. Statische Informationen werden einmal erstellt, bleiben unverändert erhalten und besitzen dokumentarischen Charakter (Aufzeichnung historischer Ereignisse, Gesetzestexte). Dynamische Informationen besitzen eine hohe Aktualität und werden in periodischen Abständen aktualisiert (Börsenkurse, Bestandsdaten). „Grundsätzlich stellen dynamische Informationen einen sehr hohen Anspruch an die Verarbeitung und Verwaltung und bedürfen einer besonderen Aufmerksamkeit.“ <ebd.> Semidynamische Informationen stellen eine Mischung aus den beiden bisher genannten Informationstypen dar. Sie besitzen einerseits einen überwiegend dokumentarischen Charakter, bedürfen jedoch unter bestimmten Umständen einer Aktualisierung (Liste von Ansprechpartnern, Preislisten).

Ein variierender Aktualisierungsbedarf erfordert nicht zwingend eine variable Handhabung der angebotenen Informationen. Bei einer großen Datenmenge sollte ein CMS jedoch verschiedene Zusatzfunktionen besitzen, um dem unterschiedlichen Anspruch der Daten entgegenzukommen. Dazu sollte z.B. eine Erinnerungsfunktion gehören, die den CMS-Benutzer periodisch an den Bedarf nach Aktualisierung erinnert. Diese Funktion könnte generell in einem Kalender mit Terminen verbunden sein. Darüber hinaus könnte ein CMS automatisch für Aktualisierungen sorgen, indem bspw. Börsendaten, Lagerbestände oder Preislisten aktuell aus einer Datenbank geladen werden.

#### **Statische Informationen**

- Aufzeichnungen historischer Ereignisse
- Biographien verstorbener Persönlichkeiten
- Hard- und Softwaredokumentationen sowie Gebrauchsanweisungen
- Betriebswirtschaftliche Dokumente
- Gesetzestexte und gerichtliche Urteile
- Auszüge aus Zeitschriften oder Büchern, Audio- und Videoaufnahmen
- Veröffentlichte Nachrichten und Pressemitteilungen

#### **Dynamische Informationen**

- Börsenkurse
- Bestandsdaten in der Lagerverwaltung
- Der Inhalt eines elektronischen Briefkastens
- Zwischenergebnisse während Sportveranstaltungen
- Aktuelle Preise bei Online-Auktionen
- Datum und Uhrzeit

#### **Semidynamische Informationen**

- Liste der Ansprechpartner eines Unternehmens
- Bibliographien und Lebensläufe noch lebender Autoren
- Preislisten, z.B. für Verbrauchsgüter

*Informationstypen und Änderungsfrequenzen von Webcontent <zsch 2002, S. 42ff.>*

MingCMS:

- Bei einer geringen Datenmenge stellt sich die Frage nach der Zweckmäßigkeit einer Unterscheidung der Inhalte und der damit verbundenen gesteigerten Komplexität des Systems. Es sollte ausreichend sein, wenn MingCMS in der Lage ist, statische Informationen in einem einfachen Arbeitsablauf für das Internet aufzubereiten. Eine Unterscheidung zwischen Informationstypen soll nur insoweit integriert sein, indem das System in der Lage ist, zu einem konfigurierbaren Zeitpunkt Daten aus einer bestimmten XML-Quelle zu laden und automatisiert darzustellen (s.u.).

### 3.1.10 Anbindung externer und interner Datenquellen

Es existieren verschiedene Möglichkeiten, einem User Daten im Internet anzubieten. Die am häufigsten verwendete ist dabei die Präsentation in HTML.-Seiten, andere Formate sind z.B. PDF-Dateien oder Flash-Seiten. Dies ist jedoch nur die Ausgabeseite. Auf der Eingabeseite liegen die Daten in deutlich mehr Formaten vor: Excel, Word, Datenbanken, XML-Dateien, Textdateien, E-Mails, etc. Die Schnittstelle, welche die Daten in eine im Web präsentierbare Form bringt, ist ein Autor. Da jedoch die Ressource eines Autors begrenzte zeitliche Möglichkeiten hat, stellt dieser gleichzeitig ein Nadelöhr dar, durch das die Daten hindurch müssen. „Der zentrale Erfolgsfaktor im Bereich Inter- und Internet besteht [jedoch] in der Aktualität der präsentierten Inhalte.“ <daib>

Die Herausforderung zur Steigerung der Aktualität besteht nun darin, die externen und internen Daten aus unterschiedlichen Datenformaten einfach anbinden zu können und über ein CMS zu verwalten. Durch den Dokumentenstandard XML vereinfacht sich die Realisierung dieses Ziels. Es ist mittlerweile möglich, aus vielen Programmen heraus eine XML-Datei mit den vorliegenden Daten zu erstellen. Wurden die XML-Daten importiert, kann durch Formatierungen eine Transformation und Selektion der Daten in das gewünschte Zielformat durchgeführt werden <ebd.>.

MingCMS:

- Wie oben unter dem Punkt „Art der Inhalte“ bereits erwähnt, soll eine Funktion integriert werden, mit der Informationen aktuell zu einem bestimmten Zeitpunkt aus einem XML-Datensatz ausgelesen werden können. Der SWF-Movie soll dann jeweils automatisch neu erstellt werden. Konnten die Daten nicht korrekt eingelesen werden, soll eine Meldung an den Benutzer gesendet werden und der alte Inhalt öffentlich bestehen bleiben.

### 3.1.11 Zwei Versionen - HTML und Flash

Viele Internetseiten, die eine Seite im SWF-Format anbieten, ermöglichen dem User auf der Startseite eine Entscheidung, ob er die Seite in einer HTML-Version oder eben einer Flash-Version betrachten möchte. Content Manager, die derzeit für Flash-Seiten angeboten werden, besitzen diese Möglichkeit meistens nicht. Dabei ist eine Trennung in zwei Versionen aus mehreren Aspekten sinnvoll. In HTML ist eine Barrierefreiheit (siehe 3.1.14) zu realisieren, in Flash dagegen bisher nicht. Zudem ist der Seitenaufbau einer reinen HTML-Seite meist deutlich schneller und zudem fühlen sich viele User durch Flash-Animationen als Intro oder für Seitenübergänge gestört. „Ein Nutzer, der [...] auf der Suche nach puren Informationen ist, wird Flash-Animationen als nervig und störend empfinden. Auch wenn diese aus Sicht des Unternehmens den Kommunikationsabsichten förderlich ist, können sie de facto die gesamte Zielsetzung der Website ins Wanken bringen. Umgekehrt kann die gleiche Technik aber auch begeistern, wenn der Nutzer Spaß und Unterhaltung sucht.“ <bors> Die einfachste Lösung bleibt folglich, den User entscheiden zu lassen, in welchem Format er die Website besuchen möchte.

MingCMS:

- Wenn der Autor es wünscht, soll automatisch eine parallele HTML-Seite erstellt werden. Dazu wird eine Startseite generiert, die als Weiche für den Besucher dienen soll (HTML- oder Flash-Version).

### 3.1.12 Flash innerhalb von HTML integriert

Für konventionelle Internetauftritte ohne eine komplett animierte Umsetzung in Flash kann eine punktuelle Integration von multimedialen Animationen sinnvoll sein. Einsatzgebiete sind z.B. statische und semidynamische Daten in Statistiken (Balkengrafiken), die ansprechend visualisiert werden sollen. Manche auf dem Markt befindliche CMS sind bereits in der Lage, Flashanimationen nicht nur zu integrieren, sondern auch zu bearbeiten. So bietet z.B. das CMS RedDot „die Möglichkeit, die Pflege multimedialer Inhalte vollständig [...] zu integrieren. Animationen können wie gewohnt bearbeitet werden.“ <redd>

MingCMS:

- Da die Seiteninhalte entweder komplett in Flash oder HTML ausgegeben werden sollen, soll eine Kombination beider Formate keine Bedeutung finden. Allerdings ist

die Unterstützung einer solchen Mischform eine weitere Anwendungsmöglichkeit für Ming (siehe 6.4).

### 3.1.13 Anzeige auf verschiedenen Ausgabegeräten

Weitere Ausgabegeräte, mit denen Inhalte einer Internetseite aufgerufen werden können, sind z.B. Handys und Personal Digital Assistants (PDA). „Damit die Informationen auch auf verschiedenen Ausgabegeräten befriedigend angezeigt werden können, ist eine komplett unterschiedliche Aufbereitung notwendig.“ <krue 2002, S. 33> Die Ursachen hierfür liegen in den Übertragungsgeschwindigkeiten, den stark variierenden Displaygrößen und der unterschiedlichen Bedienung der Geräte. Aktuelle Browser für Handys und PDAs unterstützen mittlerweile HTML, Stylesheets und die Grafikformate GIF und JPEG. Doch bereits JavaScript auf der Seite des Client, also auf dem Gerät des Users, wird herausgefiltert. Eine Unterstützung von Flash in Browsern auf mobilen Ausgabegeräten wie Handy und PDA existieren, spielen aber bisher keine große Rolle. So bietet Macromedia z.B. einen Flash Player für Pocket PCs und Handys an, mit denen z.B. ein flashbasierter Bildschirmschoner realisiert werden kann, aber eine konkrete marktorientierte Anwendung ist bislang nicht zu entdecken.

MingCMS:

- Die von MingCMS produzierten Internetseiten richten sich an User mit den am weitesten verbreiteten Browsern wie z.B. Internet Explorer, Netcape, Safari, Mozilla. Für den Autor einer Seite gilt die gleiche Anforderung – er soll seine Seiten über die gängigen Browser auf den am weitesten verbreiteten Plattformen erstellen können. Eine Anzeige auf Ausgabegeräten wie Handys und PDAs kann vorerst übergangen werden, da ein entsprechendes System zu komplex werden würde.

### 3.1.14 Barrierefreiheit

Der komplexe Bereich der Barrierefreiheit soll hier in zwei Gedanken eingeteilt werden: (1) Was sind die Muss-Anforderungen für ein Flash-CMS bei der barrierefreien Gestaltung einer Website?, und (2) Was sind die Muss-Anforderungen für ein barrierefrei bedienbares CMS?

Die für die Entwicklung von Internetinhalten existierenden Techniken und Möglichkeiten sind äußerst umfangreich, genauso wie die Mittel für den User, sich Webangebote zu betrachten. Aus diesen Gründen hat das World Wide Web Konsortium (W3C) Richtlinien rund um das Internet festgelegt. Am 5. Mai 1999 wurden diese um eine

Richtlinie erweitert „wie Web-Inhalte für Behinderte zugänglich gemacht werden können.[...] Das primäre Ziel dieser Richtlinien ist die Förderung der Zugänglichkeit.“  
<w3c> Ein CMS, das die Inhalte einer Website im Flashformat anbietet, kann bisher generell keine barrierefrei Site anbieten. Denn zum Studium einer Website benötigen Blinde z.B. eine Zusatzsoftware, welche die Texte in Laute umwandelt (einen sogenannten Screen-Reader) <voig>. Diese Software ist nicht in der Lage einen Flash-Film zu verarbeiten. Aus diesem Grund muss zur barrierefreien Gestaltung eine parallele HTML-Seite im reinen Textformat unter Berücksichtigung der oben genannten Richtlinien angeboten werden.

Soll das CMS selbst barrierefrei bedienbar sein, müssen ähnliche Kriterien beachtet werden. „Je nach Plattform für die die Clients des CMS erstellt wurden (Windows, Java, HTML, etc.) wurden hierfür bereits Guidelines durch das W3C geschaffen <ebd.>.

MingCMS:

- Durch das Problem, dass die Screen-Reader keine Flash-Inhalten verarbeiten können, muss das MingCMS zur Barrierefreiheit eine parallele HTML-Seite generieren, welche die Anforderungen des W3C erfüllt. Da die Erstellung einer solchen Parallelseite bereits in die Ansprüche an das CMS eingegangen ist, ergibt sich an dieser Stelle die Forderung nach der Erfüllung der Richtlinien zur Barrierefreiheit in der Parallelseite. Das MingCMS auch barrierefrei bedienbar zu gestalten ist eine Aufgabe mit großer Herausforderung und sollte durchaus in die konzeptionellen Überlegungen eingehen. An eine konkrete Realisierung ist an dieser Stelle jedoch nicht gedacht.

### 3.2 Bestehende Systeme und Anwendungen

Die Idee eines Flash CMS, d.h. eines Content Management Systems für animierte Internetauftritte, ist keine neue Errungenschaft. Aktuell existieren zahlreiche Systeme, die mehr oder weniger auf die Flash-Technologie zurückgreifen, sich im Einsatz befinden und mit denen auch diverse Internetauftritte realisiert wurden und werden. An dieser Stelle sollen einige Systeme und Konzepte zusammen mit den entwickelten Websites vorgestellt werden, um an Beispielen aus der Praxis zeigen zu können, auf welche Weise und zu welchen Zwecken Flash im Internet mehr oder weniger sinnvoll verwendet wird.

Allgemein scheinen bestehende Systeme für jeweils klar definierte und individuelle Aufgaben entworfen und angelegt zu sein. Einerseits finden sich Einzelseiten zur

Promotion eines Produktes oder einer künstlerischen Vereinigung, sowie komplexere, aber nicht unbedingt umfangreiche Auftritte weltweit agierender Unternehmen. Durch die Spezialisierung der Anwendungen erscheint eine Austauschbarkeit schwierig.

Allen hier beschriebenen Flash CMS ist eines gemeinsam: Die verwaltete Datenmenge ist nicht besonders groß und bleibt übersichtlich. Unterschiedlich bleiben die Arten und Verwendungen von Animationen.

- In manchen Anwendungen bleiben Animationen auf die Schaltflächen beschränkt, was den Einsatz von Flash durchaus fragwürdig macht. Schließlich lohnt sich die Verwendung eines Animationsprogramms auch nur dann, wenn es auch tatsächlich hinreichend für die Zwecke verwendet wird, für die es entwickelt wurde. Oft würde auch der Einsatz von dynamischem HTML ausreichen, um ähnliche Websites zu entwickeln, wie sie aktuell häufig mit Flash realisiert werden.
- Andere Anwendungen verwenden intensive Animationen vor allem als Preloader und für sanfte Navigations- und Inhaltsübergänge. Hier stellen sich die Unterschiede zu nicht animierten Websites deutlicher heraus: alles bleibt in Bewegung, es werden Überraschungen geboten und das Medium wird durch die permanenten Veränderungen um die Möglichkeit bereichert, eine Emotionen stärker vermitteln zu können.

Unterschiedlich ist auch die Möglichkeit zur Datenpräsentation in einer separaten HTML-Seite, was schon allein für die Barrierefreiheit unbedingt berücksichtigt werden sollte (siehe 3.1.14). Die Möglichkeiten zu einem Parallelauftritt reichen von scheinbar nicht möglich bis hervorragend realisierbar. Und schließlich findet sich auch die Möglichkeit, Flash einfach als ein Zusatzmodul in einem CMS zu integrieren, um es zur Erstellung von Unterseiten oder Einzelpräsentationen, z.B. für Vorträge, zu nutzen.

webgroove.de: Das Unternehmen webgroove.de aus Bad Salzuflen bietet ein Redaktionssystem in verschiedenen Varianten an. Das System sei komplett in Flash realisiert und biete eine vollständige und einfache Pflege von Flash Websites oder Portalen. „Bilder Texte, [...] Videos, als auch interne und externe Links lassen sich einfach und direkt einpflegen und austauschen.“ <cont3> Das Unternehmen bietet sein Flash CMS u.a. in drei Varianten unter dem Namen webpromo an, mit dem Internetauftritte zu Promotionzwecken erstellt werden sollen. Ein Basispaket bietet bspw. für 2380 Euro eine künstlerbezogenen Flashwebsite an, bei der Neuigkeiten und Termine redaktionell pflegbar sind. Dabei erstellt die Agentur die Seite und der Kunde pflegt die Inhalte ein

<webg2>. Das derzeitige Aushängeschild des Unternehmens ist die Website der Rockband 4lyn (<http://www.4lyn.de>), die größtenteils in Flash realisiert ist.

coma2: Aus München kommt das System comatool (<http://www.comatool.de>), mit dem bereits zahlreiche und umfangreiche Internetpräsenzen erstellt wurden. Es basiert auf Flash, PHP und MySQL und bietet die meisten Elementarfunktionen eines CMS (siehe Kapitel 3.1). Die Daten und Strukturen könnten ohne Kenntnisse von HTML und Flash über eine browserbasierte Schnittstelle eingegeben und aktuell gepflegt werden. Zusätzlich können auch Animationen, z.B. für Seitenübergänge erstellt werden. So verwendet bspw. die Firma Triumph International comatool, um derzeit sechzig internationale Seiten weltweit zu bearbeiten und zu aktualisieren (<http://www.triumph-international.com>). Weitere interessante Internetauftritte sind z.B. die iPAQ Virtual Tour zu einem PDA von Compaq ([http://coma2.com/ipaq/ipaq\\_show.html](http://coma2.com/ipaq/ipaq_show.html)) oder der Auftritt des Jazzduos DTS (<http://coma2.com/dts>).

contentXXL: Das System contentXXL des Nürnberger Unternehmens Portamundi (<http://www.portamundi.de>) bietet die Möglichkeit an, Mikrosites oder eine zusätzliche Flashversion zu einer realisierten HTML-Seite zu erstellen. Dabei greift das System auf den bereits bestehenden Content zurück und sei in der Lage, diesen multimedial aufzubereiten. Das CMS nutzt dabei die Möglichkeiten zur Verarbeitung von XML-Daten, die seit der Version MX vermehrt in Flash integriert sind. Die Flashseite wird von Portamundi erstellt und könne über das CMS per Browser gepflegt werden. Leider sind in den Referenzen des Unternehmens derzeit keine Websites auf der Basis von Flash zu finden.

Ego7: Eine CMS-Lösung aus den USA ist das CMSMX von Ego7 (<http://www.ego7.com>) aus New York, mit der einige sehenswerte Seiten wie z.B. der Auftritt des Schmuckherstellers Kwiat (<http://www.kwiat.com>) und eine Website für eine Kamera von Sony (<http://www.sony.jp/products/ip/movie.html>) realisiert wurden. Ähnlich wie bei contentXXL wird das Design durch die Agentur erstellt. Die Inhalte der Seiten werden über XML in die Flashseite eingelesen und können später durch den Kunden gepflegt werden.

### 3.3 Macromedia Flash

Durch den Zusammenschluss der beiden Unternehmen MacroMind und Authorware entstand im Jahre 1992 das Softwareunternehmen Macromedia mit Sitz im amerikanischen San Francisco. 1996 wurde der Vorläufer von Flash entwickelt, das Animationsprogramm FutureSplash. Es benötigt ein Plugin, um Dateien des Splash-Formats (.spl) im Browser darzustellen. Hersteller von Splash war das Unternehmen FutureWave aus San Diego, das im gleichen Jahr von Macromedia übernommen wurde. Im folgenden Jahr erschienen die Versionen 1 und 2 von Flash mit den entsprechenden Shockwave Flash Playern. Bis zur heutigen Version Flash MX 2004 wurden die Versionen 3, 4, 5 und MX veröffentlicht, wobei in der Version 4 mit ActionScript erstmals eine integrierte Programmiersprache in Flash aufgenommen wurde <kann 2002, S. 20f>. Flash von Macromedia ist inzwischen seit einigen Jahren ein Standard innerhalb der Animationstools und dazu das am weitesten im Internet verbreitete. Es existieren unterschiedlichste Anwendungen wie Internetseiten, Präsentationen, E-Learning-Tools und Benutzeroberflächen. „Mit mehr als einer Million professionellen Anwendern und einer Installationsbasis von 97 % aller internetfähigen Desktopcomputer weltweit sowie zahlreichen anderen Geräten verkörpert Flash die am weiteste verbreitete Softwareplattform der Welt.“ <macr1>

#### 3.3.1 Flash CMS-Möglichkeiten mit Macromedia-Technologien

Flashinhalte für Internetseiten werden gewöhnlich mit einer Flash-Entwicklungsumgebung produziert. Um ein Flash CMS zu realisieren wird somit auch zumindest eine Flash-Version benötigt. Derzeit existiert kein Produkt von Macromedia, das als Flash CMS bezeichnet werden könnte. Eine solches System kann allerdings z.B. mit Flash MX 2004 erstellt werden. Dabei kann bspw. eine Website entwickelt werden, welche die Inhalte aus unterschiedlichen Datenquellen bezieht und so für eine dynamische Seitengestaltung innerhalb einer animierten Umgebung sorgt. Um dynamische Inhalte innerhalb einer Flashseite darzustellen existieren mehrere Möglichkeiten:

- Einlesen von Textdateien
- Einlesen von XML-Dateien
- Anbindung an eine MySQL-Datenbank über PHP
- Einbindung externer Medien wie Bitmaps, Sounds und Video über ActionScript



Eine Erweiterung der Kombinationsmöglichkeiten ermöglicht Macromedia Flash Remoting MX. Mit diesem Werkzeug lassen sich dynamische Flashinhalte in Anwendungen integrieren, die mit Macromedia ColdFusion MX (Werkzeug zur Erstellung dynamischer Internetseiten), Microsoft .NET (Technologie, die verteilte Rechner unterstützt) oder JAVA (plattformunabhängige Programmiersprache) erstellt wurden.

Zusätzlich bietet Macromedia den Flash Communication Server an, der speziell zur Unterstützung von Streaming-Media, Audio-, Video- und Livedaten innerhalb von Flashanwendungen entwickelt wurde.

Durch eine Kombination von Flash MX und Communication Server wird die Erstellung von sehenswerten Internetanwendungen wie z.B. dem Red Bull Copilot (<http://www.redbullcopilot.com>), oder der Videoplattform Re:Evolution (<http://www.reevolution.tv>) ermöglicht. Entwickelt werden solche Anwendungen meist von spezialisierten Agenturen. Ein Anwender, der lediglich eine kleine Website erstellen möchte, wird wahrscheinlich weniger auf eine solche Technik zurückgreifen. Soll also eine kleinere dynamische und animierte Internetseite realisiert werden, kommt eigentlich nur eine aktuelle Version einer Flash-Entwicklungsumgebung, wie z.B. Macromedia Flash MX 2004, für die Entwicklung in Frage.

### 3.4 Das SWF-Format

Bei der Arbeit mit Flash muss zwischen zwei Formaten unterschieden werden. Auf der Entwicklungsseite existiert das Beschreibungsformat, das die Endung „.fla“ trägt. Das Darstellungsformat zur Anzeige trägt die Endung „.swf“ (die korrekte Aussprache lautet „swiff“). Dieses Kürzel steht für **Shockwave Flash** und hat seinen Ursprung in der Anwendung Macromedia Director.

Macromedia veröffentlichte die Spezifikationen für das SWF-Format im April des Jahres 1998. Es wurde von Beginn an angelegt, um Grafiken und Animationen effektiv über das Internet transportieren zu können. Es ist dagegen nicht dazu entworfen worden, um Grafiken zwischen Grafikprogrammen auszutauschen, wie z.B. die Formate Graphics Interchange Format (GIF) und Portable Network Graphics (PNG). Folgende Ziele sollen mit dem SWF-Format erreicht werden:

- Zur **Darstellung auf einem Bildschirm** unterstützt es Anti-Aliasing und ein schnelles Rendern von Bitmaps, Animationen und interaktiven Schaltflächen.
- Um eine hohe **Erweiterbarkeit** zu erreichen, ist das Format mit Markierungen (Tags) versehen. Dadurch soll es einfacher mit neuen Funktionen erweiterbar und kompatibel zu älteren Versionen sein.
- Dateien im SWF-Format sollen schnell über ein **Netzwerk** mit begrenzten oder unbekanntem Bandbreiten befördert werden können und sind hierfür möglichst klein komprimiert.
- Um den darstellenden Player auf der Client-Seite (Browser) möglichst klein zu halten, ist das Format auf **Einfachheit** angelegt.
- Die Dateien können **unabhängig** von externen Ressourcen, wie z.B. Schriftarten, angezeigt werden.
- SWF-Dateien weisen, u.a. durch die Darstellung mit Vektorgrafiken, eine große **Skalierbarkeit** auf.
- Das Format wurde zudem darauf ausgelegt, bei der Berechnung der Bilder eine hohe **Schnelligkeit** zu besitzen. <open>

Wie bereits erwähnt werden SWF-Dateien mit der Extension „.swf“ gespeichert und besitzen die Inhaltsbezeichnung „application/x-shockwave-flash“. Gespeichert werden sie in einem 8-bit Format und besitzen einen Header, der die Grundinformationen wie Bildrate, Größe und Version beinhaltet. Der Inhalt besteht aus den einzelnen Elementen, die durch Tags voneinander unterschieden werden. Bei der Speicherung legt das Format einen großen Wert auf Komprimierung, wobei nicht die Datei als Ganzes komprimiert wird, sondern die einzelnen Elemente <ebd.>.

### 3.5 ActionScript

Die Skriptsprache ActionScript wird dazu verwendet, Flash-Filme zu beeinflussen. Mit der Unterstützung durch ActionScript können Aktionen und Ereignisse definiert werden, sowie jede beliebige Operation vom dynamischen Laden von Animationen bis zur variablen Bewegung von Objekten und dem Ablauf sowie der Größe von Filminstanzen definiert werden <kent 2002, S. 105>. Dabei ist ActionScript eine objektorientierte Sprache (OOP), die speziell für die Animation von Websites entwickelt wurde. Allerdings handelt es sich nicht um eine eigenständige Programmiersprache, sondern vielmehr um eine Erweiterung der Entwicklungsmöglichkeiten von Flash-Filmen. „ActionScript ist immer nur in einem Flash-Film integriert wirksam und erweitert lediglich die

Operationsmöglichkeiten der jeweiligen Präsentation.“ <kann 2002, S. 20>

Eingeführt wurde die Sprache von Macromedia mit der Version Flash 3 im Jahre 1998. Fehlten hier noch wichtige Elemente einer ausgereiften Sprache, wie z.B. Arrays, wurden diese zwei Jahre später in der Version Flash 5 aufgenommen. Wichtig für die Arbeit mit dynamischen Flashinhalten wurde ab dieser Version die Integration von XML-Funktionen. Ab dieser Version wurde die Sprache auch an den Sprachstandard ECMA-262 (wird auch ECMAScript bezeichnet), definiert für Skriptsprachen, angepasst, dem auch Sprachen wie JavaScript entsprechen. Soll eine Sprache diesem Standard entsprechen, muss sie allen in der Spezifikation definierten Typen, Feldern, Objekten, Eigenschaften, Funktionen und der Programmsyntax entsprechen <ecma>. Durch diese Anpassung gelang es ActionScript relativ einfach erlernbar zu machen, wenn beim Anwender bereits Programmierkenntnisse vorhanden sind.

Unter Ming können verschiedenen Versionen von ActionScript verwendet werden. Ohne genaue Definition durch den Anwender erwartet Ming Anweisungen in der Sprachversion 5, es kann allerdings auch mitgeteilt werden, dass die Befehle nach einer früheren ActionScript-Fassung behandelt werden. Bisher weisen PHP-Dokumentationen die Ming-Funktionen (und damit auch die ActionScript-Funktionen) als experimentelle Funktionen aus, die jederzeit geändert werden können. Auch können bei weitem nicht alle Funktionen von ActionScript, wie sie in einer Flash-Entwicklungsumgebung definiert sind, verwendet werden. Existieren in Flash derzeit über 500 Sprachelemente von ActionScript <macr2>, können mit der aktuellen Version 1.0 von Ming gerade einmal knapp 50 Elemente benutzt werden. Dabei handelt es sich um absolute Basisfunktionen. Der Grund liegt in den Fähigkeiten der Ming Erweiterung, dessen Programm zur Übersetzung in SWF-Code unzureichende Fertigkeiten in der Verarbeitung von komplexen ActionScript-Funktionen aufweist. Innerhalb des Übersetzungsprogramms existiert ein hardcodierter Versatz zu den Sprunganweisungen und Programmadressen. Damit das Programm wissen kann, wohin es die Funktionswerte zurückgeben kann, sei ein hoher Aufwand an Programmmanipulation nötig <php-h2>. Jedoch können eben trotzdem einige aus ActionScript unter Flash bekannten Funktionen durch die bisher definierten Ming-Funktionen ersetzt werden.

### 3.6 PHP

Es existiert wohl kaum eine Programmiersprache, die so viele vordefinierte Funktionen zu bieten hat, wie PHP. Diese Eigenschaft macht sie zu einem mächtigen Werkzeug in der Entwicklung von Internetanwendungen, denn auf diesen Bereich ist PHP ausgerichtet. „Die erste Version entstand 1994 durch Rasmus Lerdorf, der eigentlich nur eine Möglichkeit zur Programmierung seines eigenen Webservers suchte.“ <krau 2000, S.23> Die dabei entwickelten Funktionen nannte er „Personal Home Page Tools“ und stellt diese zur freien Verbreitung ins Internet. Über die Weiterentwicklung durch Programmierer entwickelte sich die Skriptsprache PHP unter dem Namen „PHP HyperText Preprocessor“ zu der Mächtigkeit, die sie heute auszeichnet. „Zum einen wird die von ASP bekannte einfache Kombination mit HTML genutzt, zum anderen die leistungsstarke und verbreitete C-Syntax, [...] Objektorientiertes Programmieren ist möglich, ebenso wie eine gute Modularisierung.“ <ebd.> Zur weiten Verbreitung von PHP im Internet trägt u.a. die gute Anbindung an Datenbanken bei. Außerdem existieren viele eigenständige Tools zur einfachen Erstellung von PHP-Anwendungen. Und schließlich liegt der Prozessor für PHP in der Sprache C offengelegt vor, was zu einer einfachen Verbreitung, unabhängig vom verwendeten Betriebssystem, führt.

PHP wird gewöhnlich nicht als eigenständiges Programm ausgeführt, sondern direkt innerhalb von HTML-Code definiert. Wenn der PHP-Prozessor eine PHP-Datei erkennt, arbeitet er sie von oben nach unten ab. „Finden sich in der Datei PHP-Anweisungen, werden diese erkannt, ausgewertet und ausgeführt.“ <krau 2000, S. 87f>

Zur Erzeugung von SWF-Dateien wurde vor der Verbreitung von Ming bereits eine erste Bibliothek mit Funktionen erstellt. Paul Haeberli veröffentlichte diese Arbeit unter dem Titel „A C Library for Writing Flash Movies.“ <haeb> Diese SWF-Unterstützung wurde mit der Version 4.02 in PHP aufgenommen. Zur Benutzung muss wie bei Ming (siehe 3.7.1) eine Bibliothek, die LibSWF, aus dem Netz geladen und in die PHP-Umgebung des Servers kompiliert werden. Diese Funktionen ermöglichen es, einfache Objekte ebenso wie Schaltflächen zu erzeugen. Gegenüber Ming fehlen allerdings wesentliche Anforderungen: es ist kein morphen (Verwandlung von einem Objekt in ein anderes) von Objekten möglich, es unterstützt kein ActionScript, der Quellcode ist nicht öffentlich und dazu wird die LibSWF nicht weiterentwickelt <kent 2002, S. 72>.

### 3.7 Ming

Veröffentlicht wurde Ming im Januar 2000 und im Januar des folgenden Jahres als Projekt auf dem Internetportal SourceForge.net registriert. Es ist zudem als OpenSource-Bibliothek nach den Grundsätzen der GNU Lesser General Public License (LGPL) der Free Software Foundation registriert <gnus>.

Mit dieser Funktionsbibliothek wurde es möglich, mit verschiedenen Programmiersprachen SWF-Movies zu erstellen, die animierte Schaltflächen, gemorphte Objekte und die von Flash bekannten Movie Clips enthalten. „Ming is an SWF („Flash“) file format output library.“ <sour> Der Quellcode von Ming wird aktuell weiterentwickelt. Zuletzt wurde am 2. Februar 2004 die Version Ming 0.3 beta 1 veröffentlicht <ming>. Installiert werden kann Ming unter verschiedenen Systemen: Unix, Windows, MacOS X, Perl und FreeBSD.

#### 3.7.1 Ming unter PHP

Um Ming unter PHP zu nutzen, muss zunächst die dem verwendeten System entsprechende Ming-Bibliothek aus dem Internet heruntergeladen werden. Anschließend wird Ming entsprechend der mitgelieferten Dokumentation in das System installiert. Unter PHP existieren zahlreiche vordefiniert Klassen, z.B. für Formen, Bitmaps, Textfelder, Movie Clips, Tweens oder Buttons. Es ist außerdem möglich, ActionScript zu verwenden, wobei nur ungefähr 50 Funktionen definiert sind (siehe 3.5). Die Ming-Funktionen sind für PHP-Programmierer relativ einfach zu verstehen und können einfach in PHP-Skripte eingebunden werden. Durch die Kombination von Ming und PHP kann eine Flashanwendung in den Kosten reduziert werden, da keine Flash-Entwicklungsumgebung benötigt wird <kent 2002, S. 72>.

Ming, von Dave Hayden in C geschrieben, wurde mit der Version 4.05 in PHP eingeführt und ersetzte das bis dahin unterstützte Modul LibSWF.

Ein kurzes Beispiel soll zeigen, wie mit Ming unter PHP ein SWF-Movie erzeugt werden kann:

```
$movie = new SWFMovie();           //Erzeugen eines Movie-Objektes
$movie -> setDimension(500,400);   //Die Größe des Movie definieren
$movie -> setRate(25);             //Zuweisung der Bildfrequenz
```

```

$movie -> setBackground(100,100,255); //Zuweisung der
                                        Hintergrundfarbe
header("Content-type:application/x- //Datei-Header zur
shockwave-flash");                    Information des Browsers
$movie -> output();                    //Movie zur Ausgabe an den
                                        Browser senden

```

Wenn die Bibliothek korrekt eingebunden ist, kann mit diesen sechs Zeilen bereits ein vollwertiger SWF-Movie auf den Bildschirm gebracht werden.

### 3.7.2 Ming unter anderen Sprachen

Ming ist eine Bibliothek für mehrere Programmiersprachen. Zu den unterstützten interpretierten Sprachen gehören, neben PHP, auch Perl, Python und Ruby, wobei die Unterstützung für die beiden zuletzt genannten Sprachen nur unausgereift und rudimentär entwickelt ist. Die Verwendung in Kombination mit Perl ist sehr ähnlich wie mit PHP.

Unter kompilierten Sprachen kann Ming mit C++, C und Java verwendet werden. Wird bspw. C++ benutzt, wird die Bibliothek in das entsprechenden Projekt eingebunden. Die Befehle sehen sehr ähnlich aus wie unter PHP, wie dieses Beispiel des gleichen Programms wie oben unter PHP zeigen soll:

```

#include <mingpp.h> //Einbinden der Bibliothek
SWFMovie *m = new SWFMovie(); //Erzeugen eines Movie-
                               Objektes
m -> setDimension(500,400); //Die Größe des Movie
                               definieren
m -> setFrames(25); //Zuweisung der Bildfrequenz
m -> setBackground(100,100,255); //Zuweisung der
                               Hintergrundfarbe
header("Content-type:application/x- //Datei-Header zur
shockwave-flash");            Information des Browsers
m -> output(); //Movie zur Ausgabe an den
                               Browser senden

```

<ming>

## 3.8 Ming-Elemente

Wie bereits besprochen wird es durch die Ming-Bibliothek ermöglicht, mit mehreren Programmiersprachen SWF-Movies mit den dazugehörigen wesentlichen Bestandteilen zu erstellen. Ein bedeutender Teil dieser Arbeit ist es, Ming mit der Skriptsprache PHP auf seine Fähigkeiten zu untersuchen. In diesem Kapitel soll nun näher darauf eingegangen werden, ob und auf welche Weise mit PHP und Ming SWF-Elemente und komplexere Strukturen erzeugt werden können. Dabei sollen zwar alle Möglichkeiten erwähnt, aber vor allem ein intensiverer Blick auf die Elemente gerichtet werden, die für die Realisierung des angestrebten Content Management Systems mindestens nötig sind.

### 3.8.1 Erzeugung eines SWF-Movies

Im wesentlichen reichen sechs Zeilen, um einen fundierten SWF-Movie zu erzeugen (Kapitel 3.7). Die Dimension des Films spielt dabei insofern eine Rolle, da alle später hinzugefügten Elemente auf dieser Fläche Platz finden müssen. Die hier definierten Werte bestimmen die spätere Größe des Films auf dem Bildschirm. Die Bildfrequenz wird als Ganzzahl der Bilder pro Sekunde angegeben. Sie beeinflusst das Tempo, in dem der Hauptfilm abgespielt wird und ist der Hauptfaktor für die Wiedergabegeschwindigkeiten von Animationen. Und die Farbe wird schließlich in drei Werte von Rot, Grün und Blau auf einer Skala von 0 bis 255 angegeben. Wird anschließend der Header gesetzt und der Film über den Befehl „output()“ gesendet, kann der Film unter der dazugehörigen URL in einem Browser betrachtet werden. Die korrekte URL ist bei dieser Vorgehensweise die Adresse des PHP-Skriptes auf dem Webserver. Ein direkter Aufruf der Datei ohne Webserver ist wie bei allen PHP-Dateien nicht möglich, da der Server das Skript verarbeiten muss. Wie SWF-Movies generell sind auch die aus Ming erzeugten Filme so gut wie unabhängig vom verwendeten Browser.

Zur Erzeugung eines Films existiert in Ming noch eine zweite Möglichkeit. Anstatt die Datei an einen Browser zu senden, kann eine SWF-Datei erzeugt werden, die später in einem eigenständigen Flash-Player oder über ein Plug-In in einem Browser betrachtet, sowie in eine andere Anwendung eingebunden werden kann. Hierzu wird nicht der Befehl „output()“ verwendet, sondern „save()“. Der Film wird dann, wenn keine spezielle Pfadangabe gemacht wird, unter dem angegebenen Dateinamen im gleichen Verzeichnis wie das PHP-Skript gespeichert.

```
$movie -> save("ming.swf");           //Movie erzeugen und  
                                     speichern
```

### 3.8.2 Darstellung der Einzelemente

Nachdem der Film erzeugt ist, kann damit begonnen werden, verschiedenen Elemente im Film zu erzeugen und zu platzieren. Dabei eröffnet Ming Möglichkeiten zur Generierung von Text mit Textfeldern, Grafiken, Einbindung von Bitmaps, Schaltflächen und Animationen und sogenannten Tweens.

#### 3.8.2.1 Text

Für die Erzeugung und Gestaltung von Text existieren in Ming zwei Möglichkeiten. Die Klasse SWFText erzeugt einen Text zur Darstellung im Film, die Klasse SWFTextField generiert ein Textfeld, das es dem User u.a. erlaubt, selber Text einzugeben. Am einfachsten hat es sich herausgestellt, stets ein SWFTextField zu verwenden, da dieses einfacher zu handhaben ist und Konstanten zugewiesen können (z.B. das der Text nicht selektierbar ist).

Bevor allerdings ein Text ausgegeben werden kann, wird eine Schriftart benötigt. Hier beginnt ein erstes ernsthaftes Problem bei der Verwendung von Ming. Zwar können ohne größere Probleme die drei auf fast jedem System vorhandenen Schriftarten sans, serif und typewriter verwendet werden, allerdings wird die Schriftausgabe mit einem anderen Font schon schwieriger. Hierfür braucht die Ming-Klasse Font eine .fdb-Datei (Font Definition Block). Eine solche Datei enthält die komplette Definition einer Schriftart, muss aber meistens gekauft werden. Glücklicherweise haben die Entwickler von Ming dieses Problem erkannt und bieten mittlerweile einige Schriftarten zum Download bereit, die verwendet werden können, wenn die Copyright-Bedingungen eingehalten werden <ming>.

```
$font1 = new SWFFont("_sans"); //Verwendung einer Systemschrift
$font2 = new SWFFont("arial.fdb"); //Verwendung eines Schriftpakets
```

Wie bereits erwähnt ist es komfortabler, generell mit einem Textfeld zu arbeiten. Diesem können bei der Erzeugung verschiedenen Parameter zugewiesen werden. Dies wären bspw. ob das Textfeld eine Außenlinie haben soll, über mehrere Zeilen geht, der Text vom User selektiert werden kann, ob es sich um ein Passwort handelt, etc. Die folgenden Zeilen zeigen eine mögliche Variante zur Texterzeugung.

```
$text = new SWFTextField(SWF_TEXTFIELD_NOSELECT); //Erzeugung eines Textfeld-Objekts, dessen Text nicht selektierbar ist
```



```

$text -> setFont($font1);           //Zuweisung der Schriftart (Font)
$text -> setBounds(100,50);        //Definition der Außengrenzen
                                    //des Textfeldes
$text -> setColor(100,100,100);    //Farbe der Schrift im Textfeld
$text -> setHeight(12);            //Höhe der Schrift in Pixel
$text -> moveTo(100,180);         //Positionierung des Textfeldes
$text -> addString("Dies ist der Text"); //Zuweisung des Textes
$movie -> add($text);              //Text an den Hauptfilm
                                    //zuweisen

```

### 3.8.2.2 Grafik

Ming bietet die Möglichkeit, Linien, Kurven und Füllungen zu zeichnen. Natürlich ist dies nur durch Angabe von Werten innerhalb der Programmierung möglich; ein WYSIWYG ist natürlich nicht gegeben. Bei der Erzeugung von gezeichneten Objekten ist zwischen drei Objekten zu unterscheiden: SWFShape (für Linien und Kurven), SWFFILL (für Füllungen) und SWFGradient (für Füllungen mit einem Farbverlauf).

Um eine Linie oder Kurve zu zeichnen, wird eine Instanz der Klasse SWFShape erzeugt, der Linientyp definiert und eine Linie, bzw. eine Kurve gezeichnet.

```

$linie = new SWFShape() ;          //Instanz der Klasse SWFShape erzeugen
$linie ->                           //Linientyp definieren – die Argumente
setLine(1,100,100,255,200);       //sind hier Linienstärke, Rot, Grün, Blau,
                                    //Alphawert
$linie -> drawLine(100,100);      //Linie zeichnen: 100 Pixel nach rechts und
                                    //100 Pixel nach unten ausgehend von der
                                    //aktuellen Position
$linie -> drawLineTo(100,100);    //Linie zeichnen: wie oben, nur hier
                                    //ausgehend von der linken oberen Ecke
                                    //des Movies oder des aktuellen MovieClips
$linie -> movePen(50,50);         //Aktuelle Position ohne Zeichnen
                                    //verschieben. Ausgehend von letzter
                                    //Position
$linie -> movePenTo(50,50);       //Wie oben, nur hier ausgehend wie bei
                                    //drawLineTo()
$linie ->                           //Kurve zeichnen – ausgehend vom Movie
drawCurve(200,0,500,400);
$linie ->                           //Kurve zeichnen – ausgehend von der
drawCurveTo(200,0,50,400);       //aktuellen Position
$movie -> add($linie);            //Linie dem Movie zuweisen

```

Die Klasse SWFFill spielt nicht nur bei Zeichenfunktionen eine wichtige Rolle. Sie wird auch benötigt, um extern geladenen Grafiken sichtbar zu machen (siehe 3.8.2.3). An dieser Klasse zeigt sich auch eine spezielle Funktionsweise in der Handhabung von

Ming, die einem Benutzer mehrfach begegnet. Die Erzeugung eines Objektes ist zunächst mit einem übergeordneten Objekt verbunden, von dem es abgeleitet ist. Um ein Objekt vom Typ SWFFill zu erzeugen, muss zunächst die Funktion „addFill()“ auf das SWFShape-Objekt angewendet werden. Anschließend wird das entstandene Objekt dem Umriss zugewiesen.

```

$block = new SWFShape() ; //Instanz der Klasse SWFShape erzeugen
$block -> //Linientyp definieren
setLine(1,100,100,255,200);
$fill = $block -> addFill(0,155,255); //Objekt für die Füllung erzeugen. Liefert
ein Objekt vom Typ SWFFill zurück.
$block -> setRightFill($fill); oder //Zuweisung, wie das zu zeichnende
$block -> setLeftFill($fill); Objekt gefüllt werden soll. Bei
setRightFill() muss das Objekt im
Uhrzeigersinn gezeichnet werden, bei der
anderen Funktion anders herum.
$block -> movePenTo(50,50); //Zeiger in Position bringen
$block -> drawLine(100,0); //Objekt im Uhrzeigersinn zeichnen
$block -> drawLine(0,50);
$block -> drawLine(-100,0);
$block -> drawLine(0,-50);
$movie -> add($block); //Objekt an den Film zuweisen

```

Die Erzeugung eines Verlaufs funktioniert sehr ähnlich. Mit der Funktion „addEntry()“ wird die Füllung spezifiziert. Die Funktion erwartet fünf Werte: Die Stelle, an der die Farbe auftauchen soll, die RGB-Werte und einen Alpha-Wert. Gebraucht werden mindestens zwei Definitionen, damit ein Verlauf zwischen diesen beiden hergestellt werden kann.

```

$verlauf = new SWFGradient(); //Erzeugung des Verlauf-Objekts
$verlauf -> addEntry(0.0, 255,0,0); //Farbe und Position des Farbbeginns
definieren. Der Alphawert ist optional.
$verlauf -> addEntry(1.0, 0,255,0); //Zweite Farbe definieren
$fill = $block -> addFill($gradient, //Wie bei der Füllung wird der Verlauf
SWFFILL_RADIAL_GRADIENT); mit addFill() an das Objekt zugewiesen.
Der Verlauf ist ohne Zusatzangaben
linear, hier ist er radial.

```

### 3.8.2.3 Bitmap

Wie bei der Erzeugung einer Grafik bereits erwähnt, wird zur Anzeige eines Bitmaps auch eine Füllung benötigt, welche das Bild erst sichtbar werden lässt. Leider können bisher nur wenige Bildtypen verwendet werden: Bilder im Format non-progressive JPEG und Bilder im Format DBL (Define Bits Losless), zur dessen Erzeugung die Ming-

Distribution ein Tool bereit hält. Allerdings führt ein kleiner Umweg dazu, dass zumindest normale JPEGs und GIFs verwendet werden können. Diese bestens funktionierende Möglichkeit soll hier wiedergegeben werden.

```

$s = new SWFShape(); //Neues Objekt vom Typ SWFShape
$f = $s -> addFill(new //Füllung zuweisen, die auch gleich
SWFBitmap(fopen("test.jpg","rb"))); vom Typ Bitmap ist und die URL zum
Bild enthält
$s -> setRightFill($f); //dem Objekt einen Umriss zuweisen
$s -> drawLine(Breite des Bildes, 0); //Das Bild sichtbar machen
$s -> drawLine(0, Höhe des Bildes);
$s -> drawLine(-(Breite des Bildes),
0);
$s -> drawLine(0, -(Höhe des Bildes));

```

### 3.8.2.4 Schaltfläche

Wurden bereits Flächen erzeugt (siehe 3.8.2.2) ist es zu einer Schaltfläche mit Ming kein weiter Weg mehr. Es muss lediglich definiert werden, dass Grafik oder Bitmap als Schaltfläche zu behandeln sind. Ming ist auch in der Lage, Schaltflächen nach unterschiedlichen Benutzervorgängen (Überfahren, Drücken, Loslassen) zu variieren. Und damit die Schaltfläche auch ihren Zweck als Steuerungselement erfüllen kann, müssen ihr anschließend eine oder mehrere Aktionen zugewiesen werden. Dies geschieht über die Zuweisung eines ActionScript-Befehls.

```

$b = new SWFButton(); //Instanz der Klasse SWFButton
erzeugen
$b -> addShape(Form zuweisen, //Der Schaltfläche eine Form
SWFBUTTON_UP); zuweisen (siehe 4.2.2 und 4.2.3) und
einen entsprechenden Zustand
zuweisen. Hier handelt es sich um
das Loslassen der Schaltfläche durch
den User
$b -> addShape(Form zuweisen, //Zustand zuweisen: Überfahren
SWFBUTTON_OVER);
$b -> addShape(Form zuweisen, //Zustand zuweisen: Drücken
SWFBUTTON_DOWN);
$b -> addAction(new //Aktion zuweisen (siehe 4.2.7)
SWFAction("stop();"));

```

### 3.8.2.5 Animation

Heißen die Animationen in Flash Tween, so ist die Bezeichnung in Ming dagegen Morph. Auch zur Erzeugung einer solchen Animation sind bei erfolgreicher Generierung einer Fläche nur noch wenige Schritte erforderlich. Es werden zwei Objekte benötigt, eines ist das Ausgangsobjekt, das in das zweite Objekt übergeht. Dazu wird noch ein Parameter für das Tempo der Animation benötigt. Ming löst diese Herausforderung mit den Funktionen „getShape1()“, „getShape2()“ und „setRatio()“.

```
$morph = new SWFMorph();           //Instanz der Klasse SWFMorph
                                   erzeugen
$form1 = $morph -> getShape1();    //Form 1 der Animation zuweisen
$form2 = $morph -> getShape2();    //Form 2 der Animation zuweisen
$morphhandle = $movie ->         //Animation dem Movie zuweisen.,
add($morph);                       $morphhandle ist die Instanz des
                                   Objekts, mit der weiter gearbeitet
                                   wird
for($i=0; $i<10; $i++){           //Tempo der Animation. Für jeden
    $morphhandle -> setRatio($i);   Frame des Hauptfilms wird die
    $movie -> nextFrame();         Schleife einmal durchlaufen. setRatio()
}                                   gibt an, wie viele Schritte die
                                   Animation durchlaufen hat
```

### 3.8.2.6 MovieClips – Sprites

Wird eine Anwendung in Flash umfangreich, ist es gewöhnlich am geschicktesten, alle auf der Bühne auftretenden Objekte in MovieClips zu verwandeln und entsprechend zu behandeln. MovieClips bieten mehrere Vorteile: sie besitzen eine eigenen Zeitleiste, können als komplette Einheit angesprochen werden und sie können relativ problemlos angezeigt, gelöscht oder verdoppelt werden. MovieClips existieren mit all diesen Vorteilen auch in Ming, nur heißen sie hier Sprites.

Prinzipiell kann einem Sprite jedes weitere Element zugewiesen werden. Dies macht dieses Element zu einem wichtigen Teil für die Realisierung einer Navigation bzw. eines Content Management Systems. Denn Sprites können relativ einfach dazu verwendet werden, Elemente nach beliebigen Ein- und Auszublenden, in dem sie selber sichtbar oder unsichtbar geschaltet werden. Doch zunächst soll hier gezeigt werden, wie ein Sprite erzeugt wird.

```
$mc = new SWFSprite();           //Instanz der Klasse SWFSprite erzeugen
$handle = $mc -> add(Objekt);    //Dem Sprite ein Objekt zuweisen, mit
```

<code>\$handle -&gt; moveTo(x,y);</code>	der Instanz kann das Objekt weiter angesprochen werden
<code>\$mc -&gt; nextFrame();</code>	//Die Instanz positionieren (also das Objekt)
<code>\$mc -&gt; add(new SWFAction("stop();"));</code>	//Abspielkopf in der Zeitleiste um einen Frame weitersetzen, da sonst das zugewiesene Objekt noch nicht sichtbar ist.
<code>\$handle2 = \$movie -&gt; add(\$mc);</code>	//Anschließend wird der Sprite wieder gestoppt
<code>\$handle2 -&gt; setName("mc1");</code>	//Der Sprite wird dem Film zugewiesen... //...mit einem Instanznamen versehen...
<code>\$handle2 -&gt; moveTo(x,y);</code>	//...und positioniert.

So richtig interessant werden Sprites in Verbindung mit ActionScript, das im nächsten Kapitel behandelt wird.

### 3.8.2.7 ActionScript

ActionScript-Befehle können unter Ming drei Elementen zugewiesen werden: Schaltflächen, Sprites (MovieClips) und dem Movie selbst. Dabei ist zu beachten, dass nur wenige aus Flash bekannte Befehle auch in Ming verwendet werden können. (siehe 3.5) Bevor darauf eingegangen wird, wie komplexe Vorgänge mit ActionScript organisiert werden können, soll hier kurz gezeigt werden, wie eine Aktion zugewiesen wird.

<code>\$mc = new SWFSprite();</code>	//Instanz der Klasse SWFSprite erzeugen und per ActionScript (AS) anhalten.
<code>\$mc -&gt; add(new SWFAction("stop();"));</code>	
<code>\$movie = new SWFMovie();</code>	//Einen Movie erzeugen und per AS den vorher definierten Sprite verschieben.
<code>\$movie -&gt; add(new SWFAction("_root.mc1._y = 100;"));</code>	
<code>\$button = new SWFButton();</code>	//Einen Button erzeugen, der bei Betätigung per AS den vorher definierten Sprite sichtbar macht.
<code>\$button -&gt; addAction(new SWFAction("_root.mc1._visible = 1;"), SWFBUTTON_ONMOUSEDOWN);</code>	

### 3.8.2.8 Erzeugung einer automatisierten Navigation

Eine große Herausforderung stellt die Generierung einer Navigation mit Ming dar. Denn einerseits können zwar relativ einfach Elemente auf eine Bühne gebracht werden, umso schwieriger ist es dann aber, diese kontrolliert ein- und auszublenden. Bei der

Untersuchung hat sich eine Möglichkeit heraus kristallisiert, mit der diese Hürde übersprungen werden kann. Dabei werden alle Elemente, die zu einer Inhaltsseite gehören, einem bestimmten Sprite (MovieClip) zugeordnet. Wird die Seite erstmals aufgerufen, werden über ActionScript alle Sprites ausgeblendet, mit Ausnahme dessen, der beim Betreten der Seite sichtbar sein soll. Den Schaltflächen werden mehrere Aktionen zugewiesen. Zum einen sollen sie einen Sprite sichtbar machen und alle anderen unsichtbar machen. Dies ist nötig, da die Schaltfläche nicht wissen kann, welche Seite gerade aktiv ist. Aus dieser Funktionsweise ergibt sich eine große Anzahl an ActionScript-Befehlen, die diese Schaltflächen ausführen müssen. Ein exemplarischer Aufbau soll hier die Funktionsweise verdeutlichen.

Zunächst werden beim Aufruf der Seite alle Sprites unsichtbar gemacht:

```
$movie = new SWFMovie();           //Erzeugung eines Movies
$mc1 = new SWFSprite();           //Erzeugung eines MovieClips
$handle1 = $movie -> add($mc1);   //MovieClip an den Film zuweisen
$handle1 -> setName("mc1");       //MovieClip mit Instanznamen versehen
$movie -> add(new                  //MovieClip per ActionScript unsichtbar
SWFAction("_root.mc1._visible =
0;"));                             machen
```

So wird mit jedem Sprite verfahren, der später in einer Inhaltsseite auftreten soll. Die Schaltflächen erhalten dann die oben besprochene Funktionalität auf folgende Weise.

```
$b1 = new SWFButton();           //Erzeugung einer Instanz vom Typ
                                Button
for($i=0; $i < AnzahlUnterseiten; $i++){ //Die ActionScript Befehle definieren.
    $query .= "_root.mc"."$i"."_visible = //Alle MovieClips unsichtbar, sowie
0;"; //den eigenen (Die Inhaltsseite für den
} //Button) sichtbar machen.
$query .= "_root.mcX._visible = 1;";
$b1 = addAction(new             //Die Befehle dem Button zuweisen
SWFAction($query),
SWFBUTTON_MOUSEDOWN);
$movie -> add($b1);             //Den Button dem Movie zuweisen
```

Soll es ermöglicht werden, dass auf den Unterseiten zusätzlich durch mehrere Seiten geblättert werden soll, wird die Navigation entsprechend komplizierter. Allerdings kann diese gesteigerte Komplexität mit der gleichen Technik realisiert werden. Dabei müssen mehrere zusätzliche Dinge beachtet werden:

- Wenn Unterseiten vorhanden sind, muss die Schaltfläche nicht nur den MovieClip aktiv schalten, sondern auch die Schaltfläche, die für die Unternavigation zuständig ist.
- Außerdem müssen alle eventuell gerade angezeigten Schaltflächen für die Unternavigation von anderen Inhaltsseiten ausgeblendet werden.
- Jede Unterseite braucht eine eigene Schaltfläche zum vor- und zurückblättern; bei mehr als zwei Unterseiten sind für manche Seiten auch beide zusätzlichen Schaltflächen nötig.
- Alle Schaltflächen für die Unterseiten müssen ebenfalls beim Seitenstart ausgeblendet werden.

Würden diese Anforderungen fest codiert werden, wäre eine entsprechende Realisierung relativ einfach. Bei einer Generierung durch ein Content Management System müssen diese Bedingungen automatisiert erstellt werden. Damit beschäftigt sich das folgende Kapitel.

## 3.9 Open Source

Einer der Grundgedanken dieser Arbeit ist es, entstandene Lösungsansätze und Quellcodes weiterzuentwickeln, damit bestenfalls tatsächlich ein ausgereiftes Content Management System mit Ming entstehen kann. An dieser Stelle soll geklärt werden, welches die Voraussetzungen für ein Open Source Projekt sind und wo die Entwicklung veröffentlicht werden kann.

### 3.9.1 Voraussetzungen

Als Open Source gilt ein frei verfügbarer Quelltext eines Programms, zu dem zumindest eine Gemeinschaft von Entwicklern Zugang hat. Das Grundprinzip beruht darauf, dass eine größere Gruppe von Programmierern bei Bedarf direkt ein hilfreicherer und fehlerfreieres Produkt entwickeln kann, da der Quellcode offengelegt ist. Unternehmen ist es erlaubt, Open Source Software für spezielle Zwecke weiterzuentwickeln. Die bekanntesten Open Source Produkte sind der Apache Webserver, der sendmail Mailserver und das Betriebssystem Linux <tech>. Eine simple Veröffentlichung eines Quellcodes, z.B. auf der eigenen Internetseite, ist also nicht ausreichend, um ein Projekt mit Open Source betiteln zu können. Zusätzlich müssen die Lizenzbestimmungen für die Weitergabe einigen weiteren Kriterien entsprechen:

- **Freie Weitergabe:** In keinem Fall können Gebühren für eine Weitergabe verlangt werden.
- **Quellcode:** Das Programm muss den Quellcode entweder enthalten, oder es muss ein Hinweis existieren, wo der Code zum Selbstkostenpreis bezogen werden kann. Absichtlich unverständlicher Quellcode ist nicht zulässig.
- **Abgeleitete Arbeiten:** Diese müssen ohne Auflagen zugelassen sein.
- **Unversehrtheit des Autoren-Quellcodes:** Die Weitergabe von Software, die aus verändertem Quellcode hervorgegangen ist, muss erlaubt sein. Es kann jedoch verlangt werden, dass diese Software eine andere Versionsnummer trägt.
- **Keine Diskriminierung:** Die Lizenz darf niemanden benachteiligen.
- **Keine Einschränkungen des Einsatzfeldes.**
- **Weitergabe der Lizenz:** Die Rechte an einem Programm müssen auf alle Personen, die diese Software erhalten, ohne zusätzliche Lizenz übergehen.
- **Keine Einschränkung auf ein bestimmtes Produktpaket:** Wird ein Programm aus einem Paket herausgenommen, müssen für dieses die gleichen Rechte gelten wie innerhalb des Pakets.



- **Keine Einschränkung bei Weitergabe mit anderer Software:** Die Lizenz darf z.B. nicht verlangen, dass alle anderen Programme, die auf dem gleichen Medium weitergegeben werden, ebenfalls offengelegt sind.
- **Technologische Neutralität:** Die Lizenz darf keine spezielle Technologie oder einen bestimmten Stil der Schnittstelle verlangen. <opso>

### 3.9.2 Lösungsmöglichkeit

Aufgrund der Übersichtlichkeit führt die Open Source Organisation <opso3> eine Liste von Lizenzen, welche den Anforderungen der Open Source Definition entsprechen <opso2>. Eine der populärsten Gemeinschaften steht hinter dem Portal SourceForge.net <sour2>. Hier können Open Source Projekte kostenlos hinterlegt werden und stehen dadurch weltweit zur Verfügung. Durch das Einhalten der richtigen Lizenz und einer Registrierung innerhalb dieses Portals kann das Produkt dieser Arbeit offiziell als Open Source Projekt zur Verfügung gestellt werden.

## **4 Konzept für ein CMS als „echtes Autorensystem“ für die definierte Zielgruppe unter Verwendung der beschriebenen Technologien**

### **4.1 Funktionsweise des MingCMS für den Benutzer**

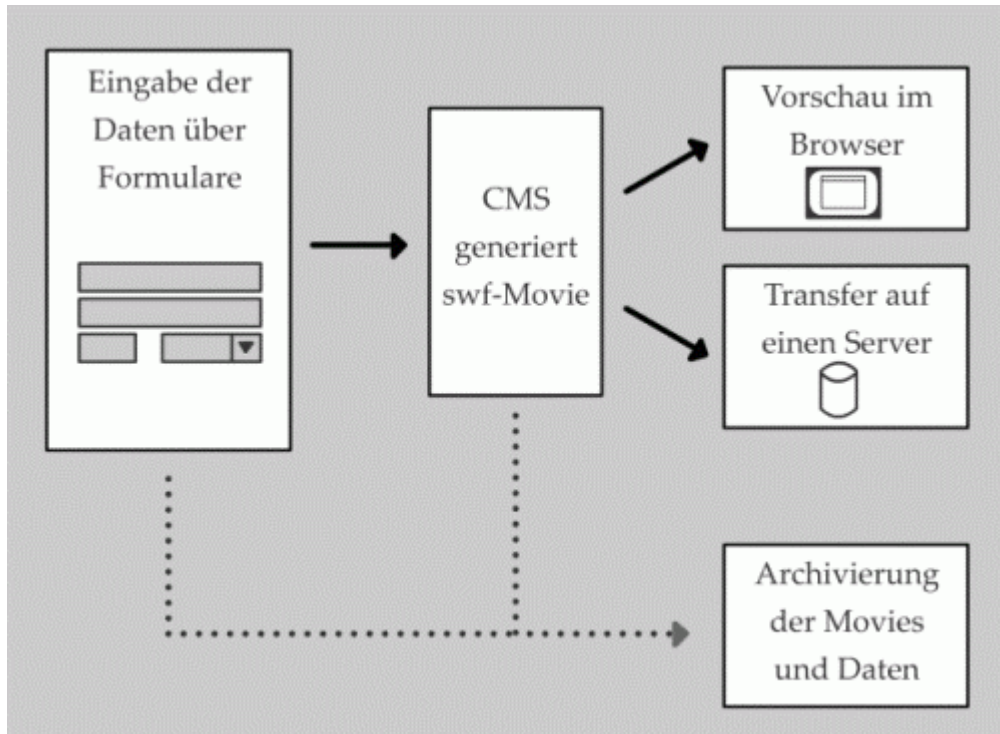
Das Ziel bleibt es, einen möglichst einfachen Arbeitsablauf – der allerdings bei jeder Software eine Grundvoraussetzung sein sollte – zu entwickeln, damit auch Benutzer, die entweder wenig Kenntnisse über Internetentwicklung besitzen oder das MingCMS nicht regelmäßig benutzen, möglichst leicht mit dem System arbeiten können.

Im Idealfall soll sich der Benutzer seine animierte Flash-Seite einrichten, indem er das MingCMS zusammen mit einem Webserver für seinen Rechner aus dem Netz lädt und beides bei sich lokal installiert. Anschließend wird MingCMS auf dem lokalen Server gestartet. Ohne Webserver ist die Anwendung nicht benutzbar, da PHP-Skripte ausgeführt werden müssen. Die Erstellung der Seite erfolgt als Dialog zwischen dem Benutzer und serverseitigen Skripten. Der Benutzer gibt Parameter an und übergibt diese an das Skript, welches wiederum die eingegebenen Daten entsprechend seiner Programmierung verarbeitet und das Ergebnis dem Benutzer als Antwort im Browser ausgibt <balz 2000, S. 946>.

Nach der Installation beginnt die Arbeit an der Website. Für den Benutzer sollen dabei gewohnte HTML-Formulare in einem Browserfenster zur Eingabe seiner Daten und Gestaltungselemente ausreichen. Befindet er sich bspw. gerade bei der Gestaltung des Hintergrundes, kann er z.B. über eine Auswahlliste zwischen einer Hintergrundfarbe und einem eigenen Hintergrundbild wählen. Anhand einer farblichen Vorlage kann der Benutzer dann z.B. die Hintergrundfarbe eingeben oder ein Bild in das System laden. Ist ein solches Einzelelement in der Gestaltung abgeschlossen, bietet das MingCMS dem Benutzer immer einen Vorschaubutton an, über den er in einem zweiten Browserfenster die bisher gestaltete Website betrachten kann.

Die fertige Seite mit dem Flash-Movie findet er während und nach der Gestaltung in einem spezifizierten Ordner. Diese Seite kann nach Abschluss der Entwicklung aus diesem Ordner kopiert und auf einen Server transferiert werden. Oder der Benutzer überträgt den SWF-Film direkt über eine Funktion im MingCMS per FTP (File Transfer Protocol) an den gewünschten öffentlichen Server. Wird eine Seite überschrieben oder

geändert, speichert das System die alten Daten in einem Archivordner.



*Arbeitsablauf für den Benutzer bei der Arbeit mit Webcontent mit MingCMS.*

## 4.2 Benutzeroberfläche

Eine optimale Benutzeroberfläche für eine Software zu entwickeln gehört wahrscheinlich zu den schwierigsten aber dabei zugleich wichtigsten Aufgaben während der Entwicklung einer Software. So können gut durchdachte Schnittstellen zu einer leichteren Interpretierbarkeit durch den Benutzer führen. Auch sind sie meist schneller zu verstehen, da einer ausgereift durchdachten Oberfläche ansehbar ist, wie sie zu bedienen sind <norm 1989, S. 12>. Diese Eigenschaften werden aktuell oft als intuitiv bezeichnet. Es gilt zu betonen, dass bei der Arbeit mit einem CMS beinahe ausschließlich über die Benutzerschnittstelle stattfindet, weswegen eine genauere Diskussion zu diesem Bindeglied von großer Wichtigkeit ist.

Schwierig wird die Aufgabe der Interfaceentwicklung dadurch, dass unterschiedliche Benutzer später das System bedienen sollen und jeder dabei die Aufgaben mit einem unterschiedlichen persönlichen Hintergrund und variierenden Fähigkeiten angeht. Ein Entwickler für Schnittstellen kann den Benutzern dadurch helfen, indem er selbst probiert ein Bewusstsein für Probleme zu entwickeln, die den Menschen bei ihrer Arbeit

begegnen könnte <rask 2001, S. 24>.

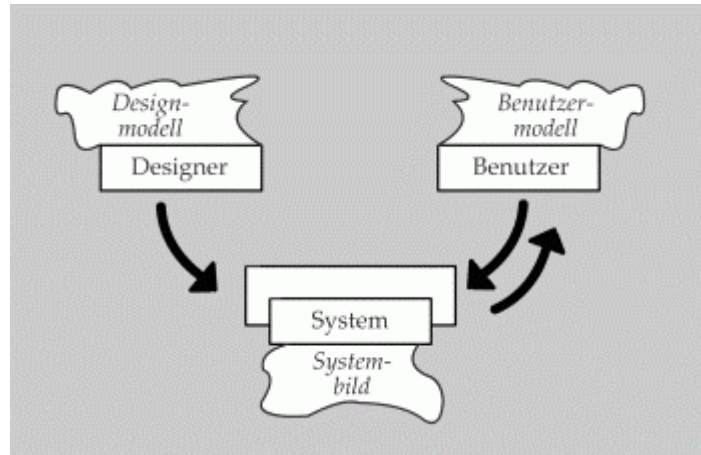
Die Entwicklung eines solchen Interfaces ist ein intensiver Prozess, innerhalb dessen zahlreiche Faktoren beachtet werden sollten. Einige Regeln zum Entwurf einer geeigneten Benutzeroberfläche, die anschließend näher erläutert werden, sind:

- Rückmeldungen durch den Benutzer
- Ein gutes konzeptuelles Modell
- Sichtbarkeit
- Mapping
- Ausrichtung auf Fehler

Als besonders erkenntnisreich gilt es, den späteren Benutzer in den Mittelpunkt der Überlegungen zu stellen, denn von dessen Zufriedenheit bei der Bedienung über ein Interface hängt teilweise der gesamte Erfolg der eigentlichen Software ab <beie 2002, S. 43ff>. Im Idealfall sollte von Beginn des Entwicklungsprozesses an ein aktueller Prototyp der Benutzeroberfläche parat stehen, an dem ein **Probepbenutzer** Aufgaben mit der Software löset. Eine solche Vorgehensweise sollte gewöhnlich dazu führen, dass der Einfluss des menschlichen Faktors – und der Benutzer ist und bleibt ein Mensch – in der Entwicklung der Oberfläche genügend Rücksicht findet. Entsprechende menschliche Faktoren sind z.B. die Aufmerksamkeitsspanne, Grenzen des Kurzzeitgedächtnisses, Lerngewohnheiten, der Grad an Erfahrung bezogen auf die Arbeit und im Umgang mit dem Computersystem sowie das mentale Modell des Benutzers von der zugrunde liegenden Struktur und dem Zweck des Systems, mit dem der Benutzer arbeiten wird <balz 2000, S. 552ff>. „Eine Schnittstelle ist dann gelungen, wenn sie auf die Bedürfnisse des Menschen eingeht und die menschlichen Schwächen berücksichtigt.“ <rask 2001, S. 24> Zudem kann zu diesen Überlegungen des Interfacedesigns die allgemein gültige Regel hinzugenommen werden, dass der Benutzer das Tempo der Interaktion, bzw. des Dialoges mit dem System, selbst bestimmen kann. Dies bedeutet, dass der Benutzer durch die Software weder warten gelassen, noch zur Eile angetrieben werden soll <ebd.>.

Weiter kann eine menschenfreundliche Benutzeroberfläche durch ein gutes **konzeptuelles Modell** und eine gute Sichtbarkeit erreicht werden. Beide Elemente ergänzen sich bei der Entwicklung einer durchdachten und klaren Bedienung. „Ein gutes konzeptuelles Modell ermöglicht es uns, die Auswirkungen unserer Handlungen vorherzusagen. Ohne ein gutes Modell gehen wir rein mechanisch, quasi blind vor.“ <norm 1989, S. 24f> Folglich ist es für den Benutzer hilfreich, wenn er jederzeit erkennen kann, welche Bedienelemente er benutzen kann und welche Wirkung eine Betätigung

nach sich ziehen wird. Eine gute **Sichtbarkeit** ist dann verwirklicht, wenn die für die Bedienung wichtigen Elemente jederzeit sichtbar sind und diese auch jeweils die richtige Botschaft an den Benutzer übermitteln <norm 1989, S. 14ff>. „Der Benutzer braucht Hilfe. Die richtige Handhabung muß ersichtlich sein, damit der Benutzer weiß, welche Teile wie funktionieren und wie er mit dem Gerät in Verbindung treten kann.“ <norm 1989, S. 18> Durch eine adäquate Sichtbarkeit kann dem Benutzer die Handhabung vereinfacht werden, da er schnell weiß, welche Möglichkeiten er besitzt.



*„Das Design-Modell ist das konzeptuelle Modell des Designers. Das Benutzer-Modell ist das konzeptuelle Modell, das sich aus der Interaktion mit dem System ergibt. Das Systembild ergibt sich aus der entstandenen [...] Struktur (einschließlich von Dokumentation, Anleitungen und Kennzeichen).“ <norm 1989, S. 28>*

Eng verwoben mit der Sichtbarkeit ist das Thema **Mapping**. Hierbei handelt es sich um einen technischen Ausdruck, „der die Beziehung zwischen zwei Dingen bezeichnet, in diesem Fall zwischen den Bedienungsvorrichtungen bzw. deren Betätigung und den realen Ergebnissen“ <norm 1989, S. 35> Durch die Benutzung von Analogien und kulturellen Standards, die meistens zu einem unmittelbaren Verständnis führen sollten, entsteht ein natürliches Mapping. Der Papierkorb auf dem Desktop erklärt sich ebenso von selbst wie ein Schieberegler mit einem blau-roten Farbbalken zur Temperatursteuerung. Ein durchdachtes Mapping erleichtert folglich die Bedienung einer Anwendung. „Wenn Dinge sichtbar sind, neigen sie dazu, einfacher zu sein, als wenn sie nicht sichtbar sind. Außerdem muß es eine enge, natürliche Beziehung zwischen der Bedienungsvorrichtung und ihrer Funktion geben: ein natürliches Mapping“ <norm 1989, S. 40>. Speziell für Benutzerschnittstellen einer computerbasierten Anwendung (z.B. ein CMS) müssen allerdings noch weitere Überlegungen zum Mapping in die Betrachtung einbezogen werden. So ist z.B. die Maus an einem Rechner ein

mittlerweile ganz natürlich verwendetes Element, mit dem intuitiv der Mauszeiger auf der Oberfläche bewegt wird. Tests aus der Vorzeit des inzwischen vertrauten Heimcomputers mit dem Bedienelement Maus zeigen allerdings, dass diese teilweise vollkommen falsch verwendet wurde (hochheben, hineinsprechen, etc.), bevor die korrekte Handhabung entdeckt wurde. Das heute jeder Computernutzer weiß, wie er die Maus zu verwenden hat, basiert letztendlich auf der Tatsache, dass sich diese Art der Bedienung durchgesetzt hat und erlernt wurde. Somit ist es nicht die Maus, die natürlich ist, sondern die gewohnte Arbeit damit. Aus diesem Grund sollte eine Benutzerschnittstelle nur sehr vorsichtig mit gängigen Modellen brechen, auch wenn diese nachweislich besser sein könnten. „Was der Kunde will, ist ein Interface mit nur wenigen Änderungen in Bezug auf die gängige Praxis – damit ist meist Microsoft Windows gemeint – das aber trotzdem irgendwie große Verbesserungen bringt. Das ist eigentlich nur in den seltensten Fällen möglich, [...]“. <task 2001, S. 178ff>

Der Umgang mit möglichen **Fehlern** sollte ebenfalls in den Entwicklungsprozess für die Benutzeroberfläche aufgenommen werden. Da jederzeit und von jedem Benutzer eine falsche Bedienung getätigt werden kann, sollte es in der Entwicklung nicht das Ziel sein, jeden möglichen Fehler zu vermeiden. Vielmehr sollte es dem Benutzer ermöglicht werden, einen Fehler wahrzunehmen und darauf entsprechend reagieren zu können. „Designer, Konstrukteure, Programmierer sollten [...] diese Richtlinien vor Augen halten:

1. Die Ursachen von Fehlern verstehen und Dinge so gestalten, dass diese Ursachen reduziert werden.
2. Die Korrektur und Revision von Handlungen ermöglichen bzw. Handlungen, die nicht rückgängig zu machen sind, erschweren.
3. Die Entdeckung von Fehlern, die dennoch vorkommen, sowie deren Behebung erleichtern.
4. Eine andere Einstellung zu Fehlern finden. Den Benutzer [...] als jemanden sehen, der eine Aufgabe zu erledigen versucht und auf dem Wege dorthin bestimmte Handlungen als unvollkommene Annäherung ausführt.“ <norm 1989, S. 157f>

Für ein CMS könnten diese Erkenntnisse bedeuten, dass z.B. beim Editieren einer Seite mehrere Schritte rückgängig gemacht werden können, um an einen Ausgangspunkt zu gelangen. Weiter könnte beim Überschreiben einer Seite darauf hingewiesen werden, ob die Seite in ein Archiv verschoben werden soll.

### 4.3 Klassische Teilung oder alles in einer Hand?

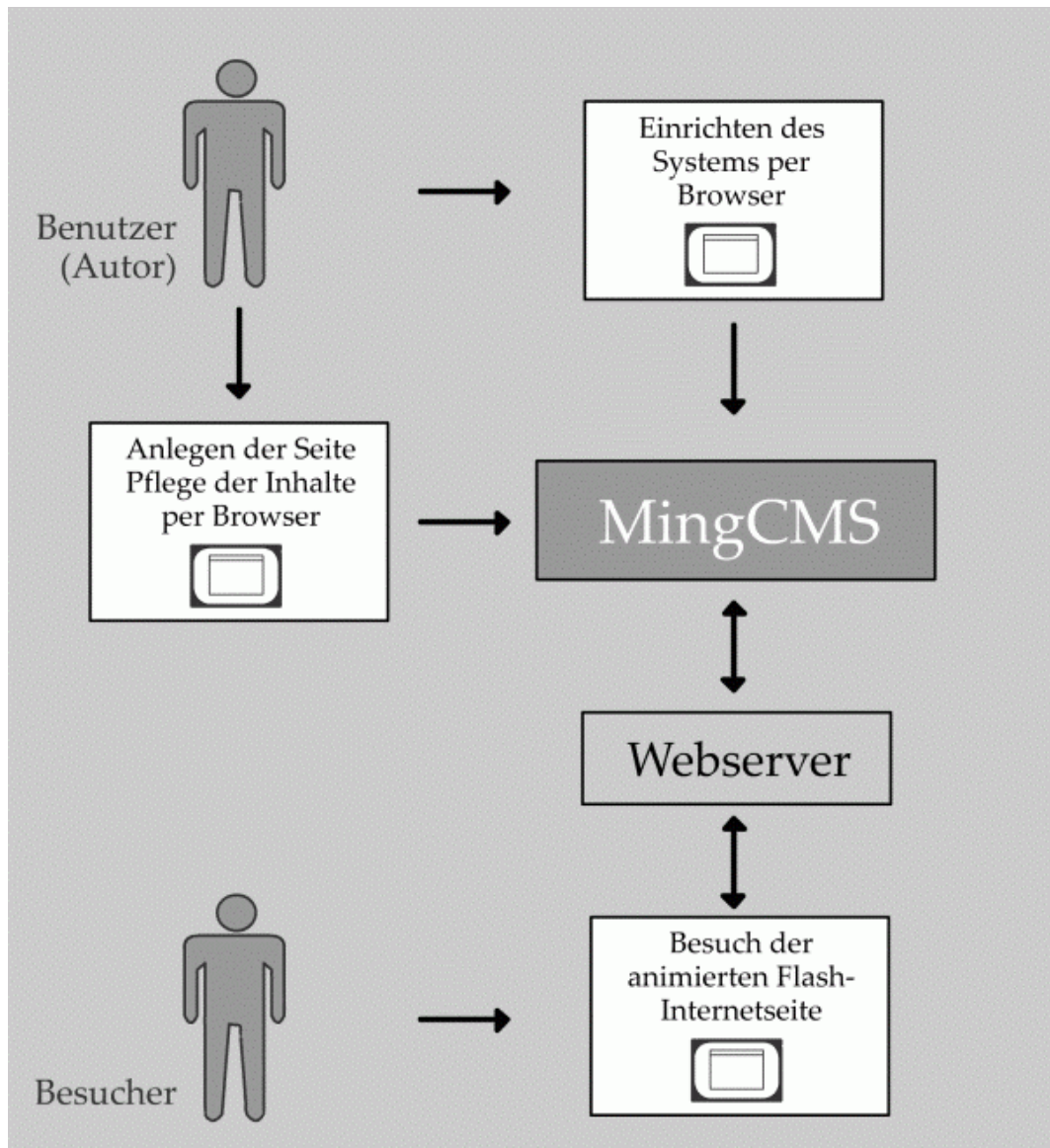
Auch im immer noch neuen und noch lange nicht ausgereiften Internet haben sich schnell Strukturen und Arbeitsabläufe etabliert. So ist es beim Web-Publishing mittlerweile zum festen Standard geworden, dass ein Auftraggeber einer Internetagentur den Auftrag übergibt, ein System einzurichten, mit dem eine eigene Redaktion aus „Internetlaien“ problemlos Inhalte erstellen und aktualisieren kann. Solche Systeme sind die Content Management Systeme. Aber ist eine solche Aufteilung zwischen Auftraggeber (Inhalte) und Agentur (Systementwicklung, Programmierung) auch immer sinnvoll?

Für komplexe Internetseiten kann diese Frage mit einem klaren „Ja“ beantwortet werden. Zu groß ist der Aufwand einer raschen und systematischen Erstellung von Internetseiten auf einer adäquaten Plattform, die aktuellen Standards entspricht, als das sie von unbedarften Internetautoren und – redakteuren erstellt werden könnte. Doch je kleiner und zielgerichteter die eigentliche Anwendung wird, desto geringer wird nicht nur der Aufwand zur Erstellung, sondern auch die Komplexität in der Einrichtung auf einer lauffähigen Plattform. Ab einer gewissen Reduzierungs- und Spezialisierungsstufe wird so der Punkt erreicht, ab dem eine Trennung von Inhalt und System nicht mehr zwingend notwendig ist. Anstatt ein klassisches Content Management System zu verwenden, genügt dem Anwender ein „echtes Autorensystem“ zur Erreichung seiner Ziele.

#### 4.3.1 Notwendigkeit zur Auflösung etablierter Strukturen für MingCMS

Aufgrund der definierten Zielgruppe, an die sich das System Ming CMS richten soll <siehe Kapitel 2.2> stellt es sich als geradezu überflüssig dar, die zur Erstellung einer Website nötigen Aufgaben von mehr als einer Person bewältigen zu lassen. Die Konzentration auf einen Ausführenden macht aus gleich mehreren Gründen Sinn:

- Enorme Kostenersparnis, da die Anwendung nicht den Weg über eine Agentur nehmen muss.
- Der komplette Überblick über die Anwendung liegt beim „Kunden“, woraus sich seine absolute Unabhängigkeit ergibt.
- Die Einrichtung des Systems und die Erstellung der Anwendung würden bei den beschriebenen Anwendungen einen meist so geringen Aufwand mit sich bringen, das der Aufwand, alle Details mit einer Agentur abzuklären, wahrscheinlich mehr Zeit in Anspruch nehmen würde, als die eigentliche Umsetzung.



*Gesamter Workflow des MingCMS: Dem Benutzer wird es ermöglicht, das System einzurichten, sowie selbstständig Seiten zu entwerfen und zu gestalten und während dem Betrieb Inhalte zu pflegen. Dies ist nicht als generelle Lösung zu verstehen, sondern entspricht den Überlegungen zu der hier durchdachten Aufgabe bei einer entsprechend definierten Zielgruppe.*

Doch auch die möglichen Nachteile sollen hier Erwähnung<sup>45</sup> finden:

- Sollte die Anwendung den Benutzer überfordern, ist der bis dahin investierte Aufwand überflüssig gewesen. Nicht zuletzt könnte sich der Benutzer durch Frustration vorerst komplett das Interesse an einem Internetauftritt verlieren.
- Hat sich die Internetseite auf dem Markt etabliert, kann ein Ausbau möglicherweise nur mit einer Agentur bewerkstelligt werden. Alle bisherigen Entwicklungen kommen dann wieder auf einen Prüfstand, den sie wahrscheinlich nicht überstehen werden.



Trotzdem zeigt es sich, dass es unter gewissen Umständen für eine genau definierte Zielgruppe sinnvoll sein kann, mit gewohnten Strukturen und Abläufen zu brechen – und sei es vielleicht nur aus experimentellen Zwecken, um zu sehen, wie sich eine Anwendung entwickelt. Wenn das gewünschte Ziel also auch auf einem anderem Weg erreicht werden kann, sich daraus mehr Vor- als Nachteile ergeben, sollte sinnigerweise ein solcher alternativer Weg begangen werden.

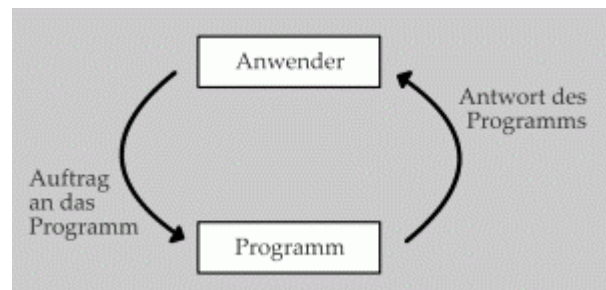
## 5 Konkrete Realisierung des MingCMS

Nachdem die verwendete Technologie diskutiert und für tauglich befunden ist, ergibt sich nun der Blick auf die praktische Realisierung. Hierbei sollen die weiteren benötigten Elemente, wie der Apache-Webserver, der Dokumentenstandard XML und MySQL-Datenbanken, einbezogen und der prinzipielle Aufbau eines Content Management Systems mit Ming konzipiert werden. Die Fähigkeiten und Grenzen eines entsprechenden Systems komplettieren dieses Kapitel.

### 5.1 Wie das Content Management System prinzipiell aufgebaut sein sollte

Nachdem in den vorangegangenen Kapiteln besprochen wurde, welche Elemente das hier zu konzipierende Content Management System enthalten und wie sich der Arbeitsablauf für den Benutzer darbieten sollte (siehe Kapitel 4), soll hier die technische Realisierung im Mittelpunkt stehen. Die hier wiedergegebenen Überlegungen orientieren sich stark an den praktischen Erfahrungen bei der Erstellung einer Internetseite mit Flash.

Prinzipiell ergibt sich eine erste Unterscheidung in der Art der zu entwickelnden Software. Das MingCMS soll zur Gestaltung einer Flash-Webanwendung benutzt werden, woraus sich zur Arbeitsweise des Programms die eines Dialogs ergibt. Hierbei übermittelt der Anwender dem



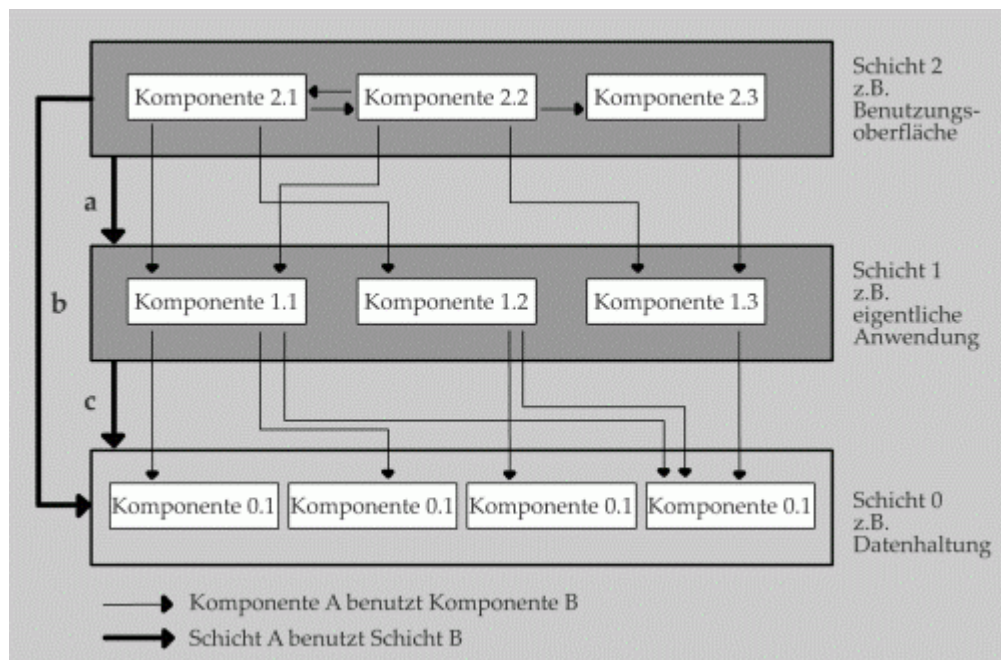
*Dialogarbeitsweise – Austausch zwischen Programm und Anwender*

Programm seine Angaben und Anforderungen, zu deren Eingabe er aufgefordert wird (Lind 1988, S. 16). Die andere Variante ist eine Software, bei der ein Stapel an Eingabedaten übermittelt wird und das Programm einen Stapel von Ausgabedaten erstellt. Diese Variante wäre allerdings durch die vielen Angaben, die vorab angegeben werden müssten, zu unfreundlich für den Benutzer.

### 5.1.1 Architektur von MingCMS

Die Konzeption eines Softwaresystems ist ein komplexer Vorgang, bei dem die einzelnen Komponenten, deren Interaktionen und die Schnittstellen klar definiert werden sollten.

„Ziel des Software-Entwurfs ist es, für das zu entwerfende Produkt eine Software-Architektur zu erstellen, die die funktionalen und nicht-funktionalen Produkthanforderungen sowie allgemeine und produktspezifische Qualitätsanforderungen erfüllt und die Schnittstellen zur Umgebung versorgt.“ <balz 2000, S. 696>



*Schichtenarchitektur – Einordnung der Elemente auf geordnete Hierarchien*

Eine gängige Strukturierung für Systemkomponenten ist eine Einteilung in verschiedenen Schichten. Die Unterscheidung der Schichten erfolgt an dieser Stelle anhand des Abstraktionsniveaus – d.h. je allgemeiner ein Element verwendbar ist, desto niedriger ist sein Abstraktionsniveau und je niedriger ist folglich seine Schicht.

- Die Grafik zeigt eine Einteilung für eine Drei-Schicht-Architektur mit einer so genannten **strikten Ordnung**. Hierbei können Schichten mit einem hohen Abstraktionsniveau auf alle niedrigeren Schichten zugreifen, aber nicht umgekehrt.
- In einer **linearen Ordnung** kann die höhere Schicht nur auf die nächstniedrigere zugreifen (Pfeil c in der Grafik wäre somit nicht erlaubt). (z.B. beim ISO/OSI-7-Schichtenmodell und dem TCP/IP-4-Schichtenarchitektur.)

- In der dritten Variante, der **baumartigen Struktur**, findet zwischen den Schichten, die sich auf der gleichen Knotenebene befinden, keine Kommunikation statt. D.h. dass z.B. Elemente der eigentlichen Anwendung in keinen Kontakt zueinander stehen <balz 2000, S. 696ff>.

Um eine geeignete Struktur für das MingCMS zu entwickeln, muss nun zunächst definiert werden, welche Systemkomponenten vorhanden sein sollen. Abgeleitet aus den bisherigen Überlegungen sind dies:

- Die Benutzeroberfläche für den Anwender,
- die anzuzeigende(n) Internetseite(n) für Seitenbesucher,
- die Eingabeaufforderungen zur Erstellung der Seiten
- die Komponente zur Generierung der Seiten
- Datenablage der aktuellen Seiten
- Archiv

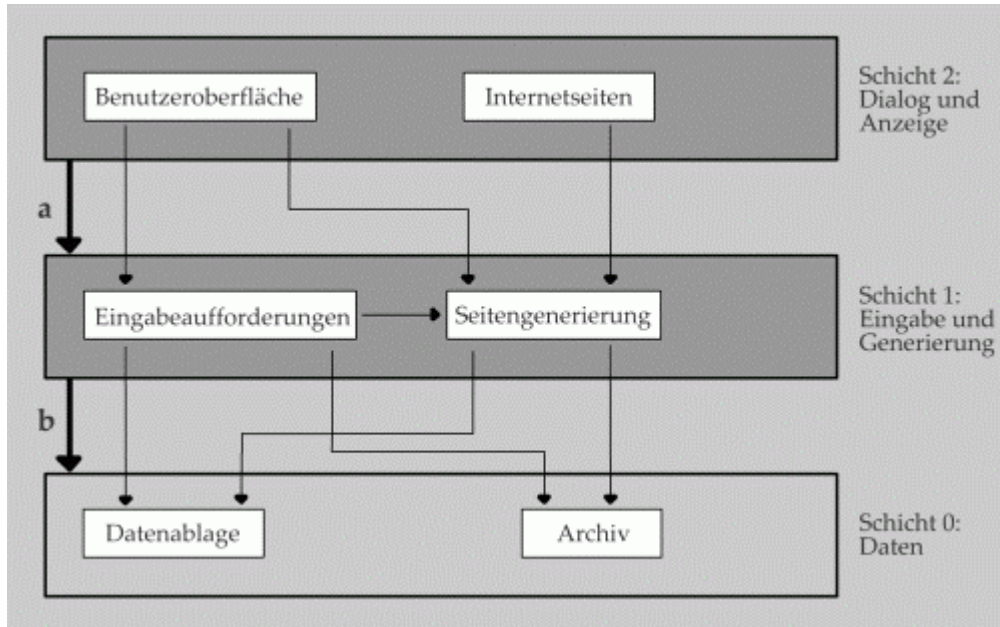
Ausgehend von der besprochenen Drei-Schicht-Architektur ist für diese CMS-Komponenten die folgende Einteilung logisch:

- Schicht 2: Benutzeroberfläche, Internetseite
- Schicht 1: Eingabeaufforderungen, Seitengenerierung
- Schicht 0: Datenablage, Archiv

Auf welche Weise müssen nun die Komponenten miteinander kommunizieren? Untereinander, also innerhalb der gleichen Schicht, sollte ein gegenseitiger Austausch zwischen der Eingabeaufforderung und der Seitengenerierung stattfinden, was eine baumartige Struktur ausschließt. Weiter ist von der Benutzeroberfläche oder der Internetseite kein Zugriff auf die Datenablage und das Archiv vorgesehen, was einer Organisation nach strikter Ordnung widerspricht. Somit ergibt sich folgende Struktur nach linearer Ordnung für das MingCMS.

Erstrebenswert ist zudem eine gute **Modularität** des Systems. Einzelne Module stellen dabei dem System individuelle Dienstleistungen zur Verfügung, auf die der Benutzer bei der Seitenerzeugung zurückgreifen kann. Solche Module könnten in einem CMS z.B. Unterstützungen zur Schaltflächenerzeugung, spezialisierte Dienste für Animationen oder die Bereitstellung von Preloadern (Informationen über den Ladestatus der Seite für den User) sein. Ein Modul besitzt zwei Elemente: Schnittstelle und Rumpf. In der Schnittstelle ist spezifiziert, welche Dienstleistungen das Modul seiner Umgebung zur

Verfügung stellt. Der Rumpf des Moduls beinhaltet die Implementierung dieser Dienstleistungen <balz 2000, S. 1050ff>. Ein solcher Aufbau würde eine Erweiterbarkeit der Fähigkeiten des Systems erhöhen, da spezialisierte Module jederzeit in das System integriert werden könnten.



Schichtenarchitektur für das MingCMS

### 5.1.2 Datenorganisation

Da eine klare Trennung zwischen der eigentlichen Anwendung und der Datenablage vorhanden sein sollte, ist es angeraten, sämtliche Daten, die Informationen über und für die Website enthalten, extern zu speichern und bei Bedarf einzubinden. Für das MingCMS werden folgende Daten benötigt:

- Datentyp 1: Angaben zur Seitenformatierung; Farben, Höhe, Breite, Animationen, Schriftart
- Datentyp 2: Seiteninhalte (Content); Texte, Bilder, Grafiken, Audiofiles, Videofiles
- Datentyp 3a: Fertige SWF-Filme
- Datentyp 3b: Archivierte SWF-Filme

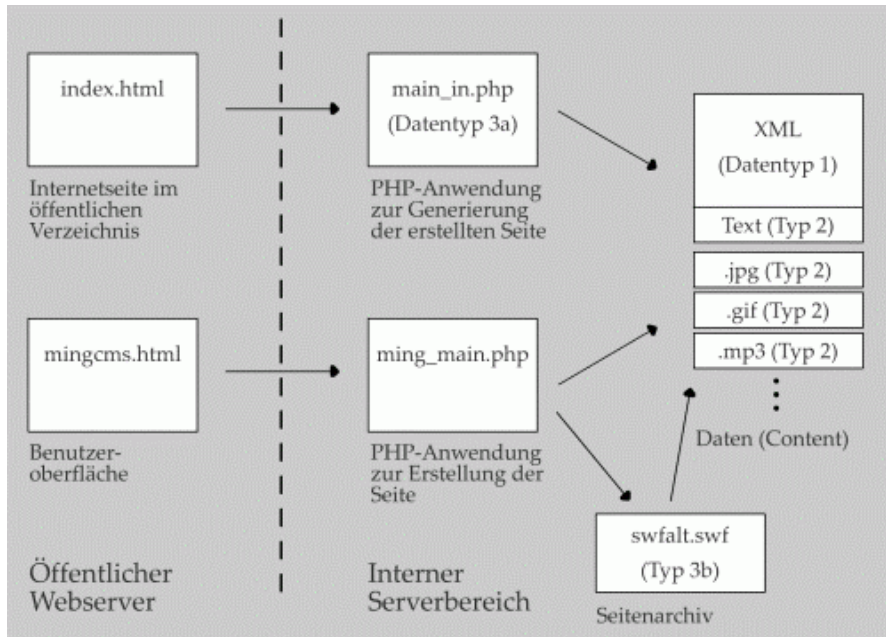
Grundsätzlich existieren bei der Verwendung von PHP mehrere Möglichkeiten zum Laden von externen Daten. Zu den populärsten Varianten gehören Textdateien, XML-Dateien und MySQL-Datenbanken. Jede dieser Varianten ist relativ einfach mit PHP zu realisieren, da es zahlreiche vordefinierte Funktionen gibt, die auf die speziellen

Anforderungen der Technik eingehen.

Welcher dieser Wege eignet sich nun für Speicherung der Daten von MingCMS, oder anders gefragt: Welche Technik eignet sich am wenigsten? Gegen eine Verwendung einer MySQL-Datenbank spricht eindeutig, dass diese zusätzlich installiert und eingerichtet werden muss. Trotz der Vorteile wie einer klarer Datenorganisation, Kompatibilität oder freier Verfügbarkeit (siehe 5.3.3) spricht dieser Nachteil gegen die Verwendung, da bei der definierten Zielgruppe nicht von einer Datenbankkenntnis ausgegangen werden kann. Für Textdateien spricht ein schneller Zugriff und die Einfachheit der Technik. Doch leider hat diese Variante einen entscheidenden Nachteil gegenüber XML: Sie ist nicht standardisiert. Bei einer Verwendung von XML ist es relativ einfach möglich, Schnittstellen zu externen Daten zu schaffen, trotzdem bleibt der Vorteil eines schnellen Zugriffs wie bei Textdateien erhalten (siehe 5.3.2). Eine Mischung aus verschiedenen Techniken empfiehlt sich in diesem Fall nicht, da das System simpel gehalten werden soll und zusätzliche Techniken diesem Ziel entgegenwirken.

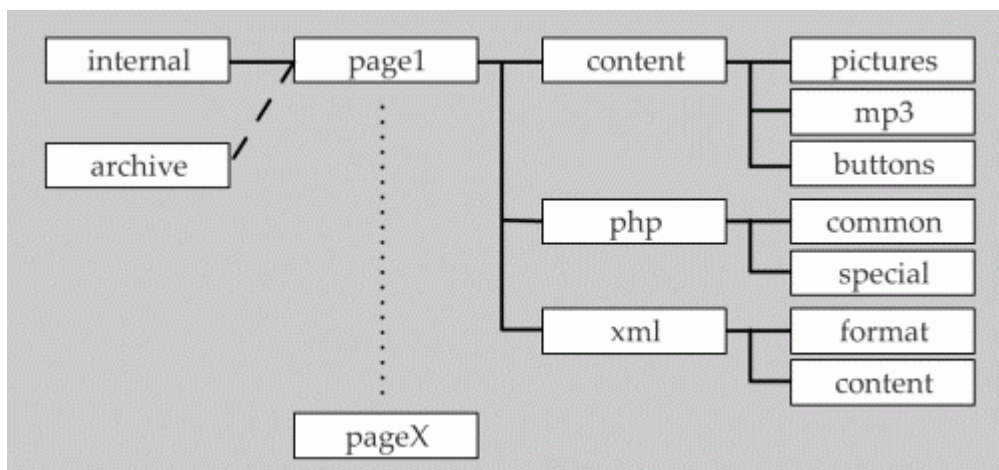
Die Daten vom Typ 1 können folglich als XML-Datensätze abgelegt werden. Für Daten vom Typ 2 drängt sich eine Mischform auf: Texte werden direkt in einem XML-Datensatz abgelegt, alle anderen Medientypen werden über eine Pfadangabe in XML spezifiziert. Die beiden Datentypen 3a und 3b sind fertige SWF-Filme, die im Archiv jeweils einen separaten Ordner erhalten sollten, in dem neben dem Film auch die jeweils dazugehörigen Dateien (.xml, .gif, .jpg, .mp3, etc.) liegen.

Zu beachten ist zudem der Aspekt der Sicherheit. Die Anwendungen werden über einen Webserver angeboten, der grundsätzlich von außen attackiert werden kann. „Die sicherste Methode ist die Platzierung der ausführbaren Dateien außerhalb der Webserver-Umgebung.“ <krau 2000, S. 80f> Diese Möglichkeit kann realisiert werden, in dem nur die Basisinternetseite im öffentlichen Webserververzeichnis liegt und alle PHP-Funktionen, XML-Dateien und externe Daten in einem separaten internen Verzeichnis angesiedelt werden. Die Basisinternetseite bindet nur die für die Ausgabe zuständige PHP-Funktion ein. Aus diesen Überlegungen ergibt sich in einer grafischen Beschreibung folgender Aufbau:



Organisation der Daten innerhalb MingCMS

Darüber hinaus sollte eine logische und übersichtliche Ordnerstruktur gefunden werden. Es empfiehlt sich, eine Trennung zwischen XML-Dateien, PHP-Dateien, Seiteninhalten und Archiv vorzunehmen. In einer tieferen Ordnung sollten zudem PHP-Dateien mit allgemeinen Funktionen von spezielleren getrennt werden. Außerdem sollte ein separater Ordner für die Archivseiten existieren, dessen Daten äquivalent zu den aktuellen Seiten aufgebaut ist. Dies könnte in etwa wie folgt aussehen.



Ordnerstruktur innerhalb MingCMS

### 5.1.3 Verarbeitung

Die Verarbeitung der Daten teilt sich in zwei Bereiche auf: Generierung der fertigen Seite auf Anfrage eines Seitenbesuchers, sowie die Erstellung der Seite durch den MingCMS-Benutzer.

#### 5.1.3.1 Erstellung der Seite

Wie unter 5.1.1 spezifiziert wurde, findet die Erstellung der animierten Internetseite in Form eines Dialoges statt. Außerdem wurde unter 4.1 festgehalten, dass die Anwendung für den MingCMS-Benutzer in einem Browser stattfinden soll, auf dessen Oberfläche je nach Schritt der Erstellung spezifische Formulare angeboten werden (zur Funktionalität der Benutzeroberfläche siehe 4.2).

Der hier angedachte technische Ablauf soll an zwei Beispielen verdeutlicht werden: (1) Erstellung des Hintergrundes sowie (2) Erstellung einer Inhaltsseite mit Text und einem Bild.

(1) Zunächst kann der Benutzer wählen, welche Elemente der Hintergrund enthalten soll: Hintergrundfarbe und Grafik. Dies kann sich über Check-Boxen vollziehen. Angenommen, er entscheidet sich für beides, bekommt er zunächst ein Eingabefeld für die Hintergrundfarbe. Anschließend kann er über ein Menü entscheiden, wie viele Hintergrundbilder er verwenden möchte. Zu jedem der Bilder wird er zur Bestimmung einer Pfadangabe aufgefordert, anschließend zur Eingabe der Position im Film.

Da sämtliche Angaben in XML-Dateien gespeichert werden, schreibt das Programm diese Dateien im Hintergrund in den entsprechenden Ordner. Diese Datei würde z.B. die Angaben Hintergrundfarbe, Bild 1 (mit Position X und Y), Bild 2 (mit Position X und Y) usw. enthalten. Ändert der Benutzer eine Angabe, wird die XML-Datei mit den neuen Parametern überschrieben. Die Vergabe der Dateinamen übernimmt das Programm.

(2) Der Benutzer muss zunächst eine Angabe machen, wie die Inhaltsseite systemintern heißen soll, damit das CMS die Verzweigungen auf der Seite kontrolliert differenzieren kann. Anschließend werden Schaltflächen benötigt. Diese werden als Grafik eingebunden, was bedeutet, dass der Benutzer zunächst einen Pfad angibt und anschließend die Position in der Navigation bestimmt. Dazu wird vom CMS abgefragt, ob die bisher vorhandenen Navigationsflächen verändert werden sollen. Ist dies geschehen, werden Position und Größe des Bereichs definiert, in dem der Inhalt



auftauchen soll. Weiter wird dann festgelegt, welchen Bereich der Text erhalten soll, der anschließend eingegeben werden kann. Schließlich wird der Bereich für das Bild definiert und eine Pfadangabe abgefragt. Auch hier schreibt das Programm im Hintergrund die entsprechenden XML-Dateien, in denen die Angaben zu den einzelnen Elementen vorhanden sind.

Die hier beschriebenen Beispiele zeigen, dass die Erstellung von Inhalten jeweils über mehrere Schritte realisiert werden. Innerhalb dieser Schritte stehen einzelne Vorgänge wie z.B. die Abfrage der Hintergrundfarbe. Die Anwendung besteht somit aus miteinander interagierenden Modulen (PHP-Skripte). Diese Module bleiben zumeist passiv und werden nur dann aktiviert, wenn ein anderes Modul eine Anfrage stellt, die eine Dienstleistung erfordert <balz 2000, S. 904>. Die einzelnen Module ordnen sich in einer Baumstruktur. Der Stamm koordiniert die Anfragen an die tiefer verzweigten Module und steuert den Ablauf des Programms.

Durch einen modularen Aufbau (siehe 5.1.1) sollte es ermöglicht werden, auf einfache Weise neue Module mit spezialisierten Dienstleistungen in das System zu integrieren. Dabei sollte das System bei seinem Start die vorhandenen Dienste einbinden und dem Benutzer deren Vorhandensein an den entsprechenden Stellen anzeigen.

Bei der Erstellung der Seiten sollte zudem eine Kopie von externen Medien hergestellt werden, wenn diese eingebunden sind. Dadurch wird erreicht, dass die für die Seite verwendeten Dateien möglichst im korrekten Verzeichnis von MingCMS liegen können und nicht irgendwo im Dateisystem des Benutzers.

### 5.1.3.2 Generierung bei Anfrage

Bei einer Anfrage durch einen Webclient (Browser) sollten bei Seiten, die unter MingCMS entwickelt wurden, zwei Möglichkeiten existieren:

- Anzeige einer statischen Seite
- Generierung einer dynamischen Seite

Handelt es sich um eine statische Seite, ist es ausreichend, den fertig erstellten Flashfilm zu übermitteln. Anders ist die Situation bei einer dynamischen Seite, die aktuell gepflegt wird oder terminabhängige Parameter berücksichtigt. Hier ruft die extern, also im öffentlichen Bereich, liegende Internetseite ein PHP-Skript, das zur Laufzeit die Flashseite generiert. Dieses „leitende“ PHP-Skript fordert, ähnlich wie bei der Erstellung der Seite,

Dienstleistungen von spezialisierten Elementen (weitere PHP-Skripte) und sorgt für die Ausgabe eines Flash-Films an den Web-Client. Die Einstellung über die Art der Seite liegt beim Produzenten der Website.

#### 5.1.4 Ausgabe/Benutzeroberfläche

Die Schicht mit der Komponente der Benutzeroberfläche, bzw. der Komponente der Ausgabe bildet von den drei Systemschichten die am einfachsten aufgebaute. Hier werden lediglich die Daten repräsentiert, die entweder vom Seitenbesucher angefordert oder vom Benutzer eingegeben wurden. Dies ist einerseits ein definierter Flashfilm, der in einer HTML-Umgebung eingebettet ist, andererseits eine Benutzeroberfläche aus HTML- und PHP-Elementen, die aus der eigentlichen Anwendung erstellt wird und einen Dialog mit dem MingCMS-Benutzer hält.

## 5.2 Notwendigkeit zur Dokumentation

Genau genommen besteht eine Software nicht nur aus den programmierten und miteinander agierenden Komponenten. Vielmehr wird eine „Einheit aus Programm und zugehöriger, das Programm beschreibender Dokumentation als Software“ bezeichnet <Lind 1988, S. 18>. Somit kann ein Programm ohne beschreibende Dokumentation genau genommen nicht als Software bezeichnet werden. Trotzdem „wird die Dokumentation im Softwareentwicklungsprozess häufig vernachlässigt. In vielen Fällen fehlt sie vollkommen. Dabei ist eine vollständige, übersichtlich gegliederte, für den jeweiligen Benutzerkreis verständliche und mit der Funktionalität des Programms übereinstimmende Dokumentation eine Grundvoraussetzung für die reibungslose Anwendung und Pflege (Änderung, Erweiterung) von Software“ <Grei>.

Eine Softwaredokumentation für das hier konzipierte CMS sollte drei Hauptfunktionen erfüllen:

- Sie sollte als **Anleitungsfunktion** dienen, die eine Grundlage für die Einsatzentscheidung bildet, sowie eine genaue Bedienungs- und Arbeitsanleitung enthalten.
- Die **Kommunikationsfunktion** soll eine einheitliche Kommunikationsbasis für Entwickler liefern, sowie die Wiederverwendbarkeit und die Transparenz, bzw. die Vergleichbarkeit, der Software erhöhen.

- Da die Software als OpenSource-Projekt angelegt sein soll, empfiehlt sich eine angemessene Dokumentation schon aus den Gründen der **Kontroll- und Nachweisfunktion** – also z.B. zur Projektfortschrittskontrolle, der Lizenzvergabe und der Transparenz gegenüber externen Prüfern <ebd.>.

Für eine Webanwendung empfiehlt sich auch die Form einer Online-Dokumentation. Jedoch ist hierbei zu beachten, dass „die Akzeptanz von Online-Dokumentationen und –hilfen [...] nur hergestellt werden [kann], wenn die Hilfetexte kurz, problemorientiert und benutzerfreundlich vernetzt sind.“ <sche> Eine Mehrsprachigkeit, so dass das Projekt mindestens zusätzlich in Englisch dokumentiert ist, sollte generell angestrebt werden.

### 5.3 Ergänzende Technologien

Zur Realisierung eines Content Management Systems mit Ming und PHP sind außer diesen beiden Technologien auch weitere Kenntnisse nötig. Schließlich soll das System auf einer Serverplattform laufen und die Daten müssen in einem Format gespeichert werden.

#### 5.3.1 Apache-Webserver

Um PHP-Skripte korrekt arbeiten zu lassen, müssen die Dateien auf einem Webserver ausgeführt werden. Der populärste und am weitesten verbreitete Webserver ist der Apache Webserver. Eine Erfassung aus dem Februar 2004 zeigt, dass 67 Prozent der weltweit aktiven Webserver von Apache sind, dem erst mit 21 Prozent Produkte von Microsoft und mit 3 Prozent Webserver von Sun Microsystems folgen <netc>. Diese Statistik und die Tatsache, dass z.B. so populäre Unternehmen und Organisationen wie Amazon (www.amazon.com), Sony (www.sony.com), Web.de (web.de), die Vereinten Nationen (www.un.org) und nicht zuletzt auch die Hochschule der Medien (www.hdm-stuttgart.de) auf einen Apache als Webserver zurückgreifen, zeigt die hohe Akzeptanz und Zuverlässigkeit dieser Software. Dazu ist ein Apache frei verfügbar, da es sich die Entwickler zum Ziel gesetzt hatten, „einen frei verfügbaren, leistungsfähigen Internetserver zu kreieren.“ <roßb 1999, S. 13> Zu den weiteren Vorteilen gehören die Unterstützung von bestehenden Standards, Authentifizierungsmechanismen zur Absicherung sensibler Bereiche, Common Gateway Interface (CGI) und Server Side Includes (SSI). Eine Verwendung zum Betrieb einer PHP-Anwendung wird ebenfalls eindrucksvoll unterstützt. Zudem arbeitet der Apache auch unter einer großen Last stabil und zügig. Und zuletzt ist der Webserver für verschiedene Plattformen wie UNIX oder

Windows erhältlich und kann über das Internet heruntergeladen werden <apac>.

Um Ming auf dem Server bereitzustellen, wird das Paket entweder aus dem Internet heruntergeladen <ming> oder es ist bereits in der PHP-Version enthalten, was ab Version 4.2.1 der Fall ist. Anschließend wird Ming für das entsprechende Betriebssystem installiert <opaq>.

### 5.3.2 XML

Die Extensible Markup Language (XML) hat sich seit ihrer Einführung vor wenigen Jahren rasch zu einem populären Dokumentenstandard entwickelt. „XML ist eine vom World Wide Web Consortium (W3C unter [www.w3c.org](http://www.w3c.org)), dem Organ, das die Standards für das Web festlegt, definierte Sprache.“ <holz 2001, S. 28> Bei XML handelt es sich um eine sogenannte Markup-Sprache, welche zur Beschreibung der Form eines Dokuments dienen. Die bekannteste dieser Sprachen ist sicherlich die Hypertext Markup Language (HTML), die zur Beschreibung von Internetseiten dient. Der Zusammenhang zwischen HTML und XML besteht darin, dass beide Sprachen auf der Standard Generalized Markup Language (SGML) basieren. XML ist wie HTML eine Teilmenge von SGML. In XML besitzt der Anwender den entscheidenden Vorteil, selbst Namen für die Markierungen (Tags) vergeben zu können. Jedoch sollte der Anwender mit dieser Möglichkeit umzugehen verstehen und sich an die wenigen, aber elementaren Regeln von XML halten können.

Die Vorteile von XML liegen darin, dass es sich als Standard etabliert hat und das Datenvolumen gegenüber der Datenmenge angemessen ist. Einige Zeilen XML beanspruchen einen um ein Vielfaches kleineren Speicherplatz als z.B. die gleichen wenigen Zeilen in Microsoft Word. Es kann mit einem einfachen Texteditor geschrieben werden und ist, bei geeigneten Markierungen, selbsterklärend. Zum Datenaustausch eignet sich XML quasi ideal, da durch die Vergabe von eigenen Markierungsnamen ein eigener Sprachstandard entwickelt werden kann. „Der Vorteil für verschiedenen Gruppen [...], ihre eigene Markup-Sprachen zu definieren, besteht darin, dass sie die Symbole und Grafiken ihrer Fachrichtungen in benutzerdefinierten Browsern verwenden können.“ <holz 2001, S. 60ff> Es existieren mittlerweile unzählige XML-Anwendungen innerhalb von Organisationen oder Interessensgruppen, die ihre Daten nach den selbst definierten Sprachstandards problemlos untereinander austauschen können. So wurden z.B. die Chemical Markup Language (CML), die Mathematical Markup Language (MathML), die Synchronized Multimedia Integration Language (SMIL) oder die Open

Software Description (OSD) mit XML entwickelt.

Dazu bietet der Sprachstandard XML einen weiteren entscheidenden Vorteil. Mittlerweile bieten Anwendungsprogramme wie Flash und Programmiersprachen wie PHP eigene Schnittstellen und Funktionen zur Anbindung an XML-Datenquellen. So ist es naheliegend, XML als Datenquelle für ein CMS zu verwenden, das unter PHP läuft. Zudem können die XML-Dateien zur Beschreibung von Seiteninhalt und –aufbau standardisiert und dadurch austauschbar gemacht werden.

### 5.3.3 MySQL-Datenbanken

Eine weitere Möglichkeit zur Speicherung von Daten ergibt sich durch die Verwendung einer Datenbank. Hierin können Daten dauerhaft und geordnet abgelegt und wiederverwendet werden. Die meisten Programmiersprachen unterstützen die Anbindung an eine Datenbank und PHP ist z.B. geradezu auf die Verwendung mit Datenbanken ausgerichtet.

Doch die meisten Datenbanksysteme sind mit hohen Kosten verbunden. Aus diesem Grund empfiehlt sich für eine OpenSource-Anwendung die Verwendung eines freien Datenbanksystems wie MySQL. „MySQL kann einschließlich seines vollständigen Quellcodes aus dem Internet oder anderen Quellen bezogen werden, ohne dass hierfür ein Preis an den Hersteller MySQL AB bezahlt werden muss. Grundlage für die Nutzung ist die GPL (GNU General Public License). Falls MySQL in kommerziellen Produkten eingesetzt wird, die nicht als Open Source weiterbetrieben werden sollen, kann auch eine kommerzielle Lizenz erworben werden.“ <thro 2002, S. 19>

Ein Datenbanksystem eignet sich in erster Linie für eine Verwendung bei datenintensiven Anwendungen. Als solche können Content Management Systeme angesehen werden, die eine große Anzahl von Seiten mit dynamischen Inhalten verwalten.

## 6 Aufwandsschätzung, Zusammenfassung und Ausblick auf weitere Verwendungsmöglichkeiten von Ming

Nach der Betrachtung der direkt und indirekt beteiligten Techniken steht die Frage an, ob das diskutierte und konzipierte System sinnvoll realisiert werden kann und was für und was gegen eine Umsetzung spricht. Außerdem sollen weitere denkbare Anwendungen für Ming diskutiert werden.

### 6.1 Realisierbarkeit des MingCMS

Die Antwort darauf, ob die konkrete Möglichkeit besteht, ein Flash-CMS mit Ming unter PHP zu realisieren, lautet definitiv Ja. Diese Beurteilung kann daran festgemacht werden, dass...

- ... es möglich ist, alle nötigen Elemente wie Texte, Grafiken, Bilder und Animationen in eine Internetseite zu integrieren,
- ... eine komplexe Navigationsstruktur realisierbar ist,
- ... PHP die Fähigkeiten mit sich bringt, um ein umfangreiches System entwickeln zu können
- ... das System auf einer einfachen Plattform aufgebaut werden kann.

Es existieren allerdings auch einige Einschränkungen, die vor einer konkreten Realisierung beachtet werden müssen.

Mit Ming ist ein Anwender bisher nicht in der Lage, die meisten aus Flash bekannten ActionScript-Funktionen zu realisieren. (siehe 6.3)

Eine Skriptsprache wie PHP ist ursprünglich nicht daraus ausgerichtet, als Sprache zur Entwicklung einer komplexen Software zu dienen. So ist z.B. die Sicherheit durch eine Vergabe von definierten Datentypen ein wichtiger Bestandteil von umfangreichen Programmen und kann durch Skriptsprachen meistens nicht realisiert werden <balz 2000, S. 947f>. Glücklicherweise bildet aber PHP unter den Skriptsprachen eine Ausnahme in der Funktionalität. „Für die serverseitige Programmierung stehen dem Programmierer umfangreiche Funktionsbibliotheken zur Verfügung, in denen ziemlich alles enthalten ist, was für die Webprogrammierung benötigt wird.“ <dehn 2001, S. 316> Somit dürfte zumindest ein CMS, das für die Bereitstellung einer übersichtlichen Menge von Internetinhalten zuständig ist, mit PHP realisierbar sein. Allerdings sollte berücksichtigt

werden, dass Ming mit den kompilierten Sprachen C und C++ ebenso umfangreich ist wie unter PHP. Und eine kompilierte Sprache ist in der Schnelligkeit um ein vielfaches leistungsfähiger als eine interpretierte Sprache wie PHP.

### 6.1.1 Aufwandsschätzung

Der Aufwand der praktischen Umsetzung ist nicht zu unterschätzen. Es müssen zahlreiche Funktionen geschrieben, eine ausgefeilte Benutzerführung entwickelt und eine ausgiebige Dokumentation erstellt werden. Zur Einschätzung des Entwicklungsaufwands existieren verschiedene Methoden. Ein wichtiger Faktor in der Berechnung ist die geschätzte Anzahl von Programmzeilen, den Lines of Code (LOC). Dazu gibt es die Kenngrößen von Mitarbeiterjahren (MJ) und Mitarbeitermonaten (MM) bzw. Personenmonaten (PM).

Eine Untersuchung der Firma Hewlett-Packard zu insgesamt 135 realisierten Projekten ergab folgende Faustregel: „Eine durchschnittliche Software-Entwicklung liefert ungefähr 350 Quellcodezeilen (ohne Kommentare) pro Ingenieurmonat. Dabei umfasst die Ingenieurzeit alle Phasen von der Definition bis zur Implementierung.“ <balz 2000, S. 74ff> Die Entwicklungsdauer wird dadurch beeinflusst, dass bei einer kürzeren Entwicklungszeit mehr Mitarbeiter benötigt werden, deren Produktivität aber dadurch sinkt, da der Kommunikationsaufwand der Mitarbeiter untereinander steigt. Ist die Entwicklungszeit länger, sinkt der Kommunikationsaufwand bei der dann geringeren Anzahl an Mitarbeitern und die Produktivität steigt. Folgende Formel berücksichtigt diese Faktoren:

$$\text{Optimale Entwicklungsdauer} = 2,5 * (\text{Aufwand in MM})^s$$

mit  $s = 0,38$  für Stapel-Systeme

$s = 0,35$  für Dialog-Systeme

$s = 0,32$  für Echtzeit-Systeme < ebd.>

Wie viele Zeilen könnte ein Basissystem des MingCMS benötigen? Eine vorsichtige und großzügige Schätzung ergibt folgende Überlegung: Für eine einfache Internetseite mit Ming werden 500 Zeilen Code benötigt (die nicht in die Berechnung eingehen, da sie vom System erzeugt werden), diese Zeilen müssen durch die Software generiert werden (1000 Zeilen), die Benutzerschnittstelle und die Komponente zur Erstellung der Seite sind die größten Teile (2000 Zeilen), den Rest teilen sich die Datenverwaltung und das Archiv (je 500 Zeilen).

Für MingCMS ergibt sich aus dieser vorsichtigen Schätzung, dass für ein Basissystem etwa 4000 Programmzeilen benötigt werden und es sich bei MingCMS um ein Dialog-System handelt, folgende Berechnung:

$4000 \text{ LOC} / 350 \text{ Zeilen pro Monat} = 11,4 \text{ MM (Mitarbeitermonate)}$

$\text{Optimale Entwicklungsdauer} = 2,5 * (11,4)^{0,35} = 5,9 \text{ Monate}$

$\text{Durchschnittliche Mitarbeiterzahl} = 11,4 \text{ MM} / 5,9 \text{ Monate} = 1,9 \approx 2 \text{ Mitarbeiter}$

Somit ergibt sich ein Richtwert, dass zwei Mitarbeiter ziemlich genau ein halbes Jahr zur Realisierung eines Flash-CMS mit Ming und PHP benötigen würden. Da das System auf dem Open-Source-Gedanken basieren soll, ist jedoch auch die Möglichkeit vorhanden, dass die Entwicklung ab einem gewissen Stadium zu einem Selbstläufer werden kann.

## 6.2 Grenzen in der Erweiterbarkeit

Eingeschränkt wird die Erweiterbarkeit, d.h. einer Vergrößerung der Fähigkeiten des Programms, eines Flash-CMS auf Ming/PHP-Basis durch mehrere Faktoren:

Geschwindigkeit der Seitengenerierung, Datenorganisation, verwendete Sprache, Ming-Sprachelemente.

Die Generierung einer übersichtlichen Internetseite mit einigen Grafiken innerhalb von zehn Unterseiten erzeugt einen Flash-Movie mit einer Dateigröße, die schnell 100 Kilobyte erreicht. Die **Schnelligkeit in der Erzeugung** wird durch mehrere Dinge ausgebremst. Zum einen müssen die PHP-Funktionen mehrfach auf das Dateisystem zurückgreifen, z.B. bei der Einbindung eines Bildes oder beim Auslesen einer XML-Quelldatei. Dieser Vorgang vollzieht sich in einem einzelnen Vorgang sehr schnell; stapeln sich jedoch die Anfragen, so ist der Zeitverbrauch entsprechend hoch. Zum anderen wird ein PHP-Skript jedes Mal aufs Neue von oben nach unten abgearbeitet, was das Tempo des Programms reduziert. Um das Programm möglichst modular zu halten, müssen Funktionen ausgelagert werden. Die Einbindung dieser externen Funktionen ist ein zusätzliches bremsendes Element. Wenn die Anwendung in der Komplexität wächst, wird sich dies nicht gravierend, aber merklich in der Geschwindigkeit der Seitengenerierung bemerkbar machen. Dies sind alles Probleme, die durch die Wahl der **Sprache** PHP entstehen. Bei einer Anwendung in C oder C++ werden diese Schwierigkeiten reduziert (siehe 6.1).

Die **Datenorganisation** ist sicherlich ein Problem, das durch eine intelligent angelegte Struktur in den Griff bekommen werden kann. Es ist jedoch nicht zu vernachlässigen,



dass schnell Dutzende von XML-Dateien zusammenkommen, in denen die Daten abgespeichert sind. Strotzt eine Seite vor Animationen und Grafiken, muss auch hierfür eine geeignete Organisation gefunden werden.

Die Sprachelemente von Ming bringen für die Erweiterbarkeit das entscheidende Problem mit sich, dass zahlreiche Möglichkeiten bisher nicht vorhanden sind. Zur Lösung vieler Probleme müssen Umwege über PHP programmiert werden, die natürlich einen höheren Programmieraufwand nach sich ziehen und für einen längeren Verarbeitungsprozess sorgen.

### 6.3 Fähigkeiten und Grenzen von Ming

Einerseits begeistert Ming auf den ersten Blick. Es ist, wie bereits erwähnt, relativ leicht möglich, mit der Verwendung einer Sprache wie PHP ausgefeilte Flashanwendungen zu erzeugen – die Elemente sind schnell animiert und kombiniert. Doch die Ernüchterung hält bei der Entwicklung mit Ming auch schnell Einzug, da viele Anwendungen nur schwierig oder gar nicht zu realisieren sind.

Zudem gilt nach wie vor, dass Ming innerhalb von PHP den Status „experimentell“ trägt. Abgelöst hat Ming eine SWF-Erweiterung in PHP, die den Namen libswf trug. Diese wurde 1999 von Paul Haeberli entwickelt, aber nicht weiter ausgebaut. Gegenüber dieser Bibliothek hat Ming derzeit den Vorteil, dass es weiterentwickelt wird. Es ist jedoch durchaus möglich, dass Ming in einer neueren Version von PHP nicht mehr integriert ist und durch eine andere SWF-Erweiterung ersetzt wird. Bis dahin gilt vorerst: „Use this extension at your own risk.“ <phpn>

#### 6.3.1 Handhabung und Organisation von Elementen und Daten

Es ist in Ming nur auf kompliziertem Wege möglich, verschiedene Schriftarten zu verwenden. Es stehen nur drei Systemschriften zur Verfügung und zusätzliche Schriftarten müssen separat erstellt oder, teilweise mit Kosten verbunden, aus dem Netz geladen werden. Die Einbindung unterschiedlicher Grafikformate, mit Ausnahme von PNG und JPEG, ist ebenfalls nur über eine Konvertierung der Dateien möglich. Immerhin sind die Entwickler von Ming derzeit um eine Abhilfe und Erweiterungen bemüht. Weiter gehört es zu der Liste von Schwierigkeiten, dass bisher die zahlreichen Funktionen, die von ActionScript innerhalb der Flash-Entwicklungsumgebung bekannt sind, innerhalb von Ming nur rudimentär benutzt werden können. Und zuletzt bringt die

Arbeit mit Ming das Problem mit sich, dass kaum Dokumentationen vorhanden sind. Diese werden aber dringend benötigt, da in der Entwicklung immer wieder Funktionen nicht auf die gewünschte Weise funktionieren und kein Anhaltspunkt darüber existiert, wo der Fehler liegt. Die Fehlermeldungen sind sehr dürftig und bringen einen Entwickler nur selten auf die richtige Fährte zum Fehler. Meistens ist es auch der Fall, das überhaupt keine Ausgabe stattfindet, wenn irgendwo in der Flasherzeugung ein Fehler steckt. Der Bildschirm bleibt dann einfach leer.

Dies ist es aber eigentlich schon mit den weniger ausgereiften Eigenschaften von Ming. Ansonsten bringt die Erweiterung alle Basisfunktionen mit sich, um ein Flashanwendung zu realisieren. Alle dafür nötigen Elemente können erzeugt, gehandhabt und gelöscht werden. Eine gezielte Steuerung der Elemente ist ebenfalls möglich. Die aus Flash bekannten MovieClips können genau wie in Flash mit allen Zusätzen (eigene Zeitleiste, separate Steuerung) verwendet werden.

Praktisch ist zudem die Tatsache, dass es möglich ist, Daten aus Ming an HTML-Seiten zu übergeben. So können z.B. Benutzerdaten aus einem Textfeld weiter verarbeitet werden, sei es in HTML-Seiten oder in einem PHP-Skript.

### 6.3.2 Animationsmöglichkeiten

Die Animation wird hier in zwei Bereiche eingeteilt:

- Animation als Bewegung von Objekten (Animation) – in Flash: Bewegungstween.
- Animation als Verwandlung von einem Objekt in ein anderes (Morphing) – in Flash: Formtween.

Hinzu kommt noch eine dritte Variante von animierten Objekten: Schaltflächen, die bei Mausbewegungen Veränderungen zeigen.

Alle Varianten sind mit Ming realisierbar, es existieren jedoch, wie auch in Flash von Macromedia, Einschränkungen. Die Bewegung von Objekten ist problemlos anwendbar. In Ming gezeichnete Formen können bewegt und gedreht werden. Das gleiche gilt für Grafikdateien wie Bitmaps.

Die Verwandlung von einem Objekt in ein anderes ist für in Ming erzeugte Formen ebenfalls relativ problemlos realisierbar. Formen können gezogen, gestaucht, skaliert werden oder eben komplett in eine andere Form übergehen. Als Einschränkung gilt

jedoch, dass keine Grafikdateien für ein Morphing verwendet werden können. Das gleiche gilt aber parallel auch in Macromedia Flash, wobei hier zumindest über Umwege eine entsprechende Verwandlung möglich wird.

Der Funktionsumfang für Schaltflächen ist hervorragend. Ming kann verschiedene Zustände, abhängig von Mausbewegung und -aktivität, verwalten. Hierbei ist es sowohl möglich, verschiedene in Ming erzeugte Formen zu verwenden, als auch fertige Grafikdateien einzubinden.

#### 6.4 Zusätzliche Anwendungsmöglichkeiten für Ming

Die Realisierung eines FlashCMS mit Ming ist eine umfangreiche Aufgabe. Natürlich zeigen sich hieran exemplarisch sehr gut die Möglichkeiten und die Grenzen dieser Technik. Aber Ming könnte durchaus für Anwendungen verwendet werden, die spezifischer auf eine Aufgabe zugeschnitten sind.

Eine der wenigen Möglichkeiten, im Internet Geld zu verdienen, sind **Werbebanner**. Es existieren unterschiedliche Größen und Formen und viele der Werbebanner sind in Flash realisiert. Dies ist auch naheliegend, da Werbebanner auffallen sollen und dies mit animierten Elementen leichter möglich ist als mit statischen Mitteln. Eine mögliche Anwendung für Ming wäre die Erstellung und Verwaltung von Flash-Werbebanner – die natürlich zusätzlich auch als statisches Bild generiert werden sollten. Ein solches System könnte dafür sorgen, dass eine animierte Grafik erstellt werden kann, oder auf ein fertiges Bild eine animierte Typographie gesetzt wird, oder auch Elemente ansprechend animiert werden können. Auch könnte der User direkt im Werbebanner zur Eingabe von Daten aufgefordert werden, mit denen bei einer Weiterleitung auf die beworbene Seite eine personalisierte Seite aufgerufen wird. Die verschiedenen Größen und Formen von Werbebanner könnten automatisch skaliert und produziert werden. Eine Content-Verwaltung könnte z.B. dafür sorgen, dass bei jedem neuen Seitenaufruf ein anderer Slogan in den gleichen Werbebanner geladen und angezeigt wird.

Weiter eignet sich Ming zur Erstellung **kleinerer Animationen**. Wenn für eine Anwendung eine einfache Animation benötigt wird, kann eine Umsetzung in einer der genannten Programmiersprachen durchaus rentabler sein als die mit dem Kauf von Macromedia Flash verbundenen Kosten. Eine Software zur Erstellung von Animationen mit Ming ist wohl zu aufwendig, so dass ein solches System sicher nicht vorteilhaft ist.

Da es möglich ist, aus Ming heraus Daten an HTML-Seiten weiterzuleiten, könnte diese Technik auch zum Aufbau einer animierten **Navigation** innerhalb einer HTML-Internetseite verwendet werden.

Weitere Anwendungsmöglichkeiten könnten außerdem sein: <kent 2002, S. 71>

- Schlagzeilen, die in Laufschrift animiert werden sollen.
- Einfache Online-Spiele
- Online-CAD, wie z.B. auf der Internetseite von Lego, wo Online Legowelten gebaut werden können <lego>.
- Durch die Möglichkeit MP3-Dateien zu streamen könnte Ming auch zur Erstellung einer Audioanwendung über das Internet mit einer SWF-Schnittstelle verwendet werden. <jarg>

Sicherlich spielt bei der Anwendung von Ming auch das Interesse, die Phantasie und die Fähigkeiten des Programmierers eine große Rolle.

## Literaturverzeichnis

**Bücher, Artikel und Internetquellen sind gemeinsam aufgeführt.**

- apac      The Apache Software Foundation »URL: <http://www.apache.org> [zuletzt besucht am 23.03.2004]«.
- balz      Balzert, Helmut: Lehrbuch der Software-Technik – Software-Entwicklung, Spektrum Akademischer Verlag GmbH, Heidelberg/Berlin 2000.
- beie      Beier, Markus/Gizycki, Vittoria von: Usablity – Nutzerfreundliches Web-Design, Springer, Berlin 2002.
- bors      Borsutzky, Silvana »URL: <http://www.contentmanager.de/itguide/marktuebersicht.html> [zuletzt besucht am 01.03.2004]«.
- cont1     contentmanager.de »URL: <http://www.contentmanager.de> [zuletzt besucht am 01.03.2004]«.
- cont2     contentmanager.de »URL: <http://www.contentmanager.de/itguide/marktuebersicht.html> [zuletzt besucht am 01.03.2004]«.
- cont3     contentmanager.de »URL: [http://www.contentmanager.de/itguide/produkt\\_1235\\_flash\\_cms.html](http://www.contentmanager.de/itguide/produkt_1235_flash_cms.html) [zuletzt besucht am 09.03.2004]«.
- dehn      Dehnhardt, Wolfgang: Scriptsprachen für dynamische Webauftritte, Carl Hanser Verlag, München/Wien 2001.
- daib      Daiberl, Jürgen »URL: [http://www.contentmanager.de/magazin/artikel\\_87\\_anbindung\\_interner\\_und\\_externer\\_datenquellen\\_an.html](http://www.contentmanager.de/magazin/artikel_87_anbindung_interner_und_externer_datenquellen_an.html) [zuletzt besucht am 09.03.2004]«.
- ecma      Ecma International »URL: <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf> [zuletzt besucht am 15.03.2004]«.
- gnus      GNU's Not Unix! »URL: <http://www.gnu.org/copyleft/lesser.txt> [zuletzt besucht am 16.03.2004]«.

- grei Greif, Norbert/Schrepf, Heike »URL: <http://www.berlin.ptb.de/8/83/831/swq/doc/drl/drl.pdf> [zuletzt besucht am 23.03.2004]«.
- haeb Haeberli, Paul »URL: <ftp://ftp.sgi.com/cgi/graphics/grafica/flash/DOC.html> [zuletzt besucht am 15.03.2004]«.
- holz Holzner, Steven: XML, Markt+Technik Verlag, München 2001.
- jarg Jargon francais »URL: <http://www.teleferique.org/stations/Erational/jargon/> [zuletzt besucht am 30.03.2004]«.
- kann Kannengießer, Matthias: ActionScript – Das Praxisbuch, Franzis Verlag GmbH, Poing 2002.
- kent Kent, Allan/O’Dell, Devon H./Chase, Andy/Rosa, Jan/Abraham, Sanjay/Suyoto, Iman S.H./Apshankar, Kapil: Professional PHP4 Multimedia Programming, Wrox PressLtd., Birmingham (UK) 2002.
- krau Krause, Jörg: PHP – Webserver-Programmierung unter Windows und Linux, Carl Hanser Verlag, München/Wien 2000.
- krue Krüger, Jörg Dennis/Kopp, Matthias: Web Content managen – Professioneller Einsatz von Content-Management-Systemen, Markt + Technik Verlag, München 2002.
- lego LEGO.com »URL: <http://www.lego.com/eng/create/activities/worldbuilder/> [zuletzt besucht am 30.03.2004]«.
- lind Lindner, Ulrich/Trautloft, Rainer: Grundlagen der problemorientierten Programmentwicklung, VEB Verlag Technik, Berlin 1988.
- macr1 Macromedia »URL: <http://www.macromedia.com/de/software/flash/productinfo/overview/> [zuletzt besucht am 08.03.2004]«.
- macr2 Macromedia »URL: [http://www.macromedia.com/support/flash/action\\_scripts/actionscript\\_dictionary/](http://www.macromedia.com/support/flash/action_scripts/actionscript_dictionary/) [zuletzt besucht am 15.03.2004]«.
- ming Ming »URL: <http://ming.sourceforge.net/index.html> [zuletzt besucht am 15.03.2004]«.

- netc Netcraft »URL: <http://news.netcraft.com/archives/2004/02/index.html> [zuletzt besucht am 23.03.2004]« [zuletzt besucht am 15.03.2004]«.
- norm Norman, Donald A.: Dinge des Alltags – Gutes Design und Psychologie für Gebrauchsgegenstände, Campus Verlag, Frankfurt 1989.
- opaq OpaqueWiki::MingInstall »<http://www.opaque.net/wiki/index.php?MingInstall> [zuletzt besucht am 30.03.2004]«.
- open OpenSWF.org »URL: <http://www.openswf.org/spec/SWFfileformat.html> [zuletzt besucht am 15.03.2004]«.
- opso1 open source »URL: <http://www.opensource.org/docs/definition.php> [zuletzt besucht am 16.03.2004]«.
- opso2 open source »URL: <http://www.opensource.org/licenses/> [zuletzt besucht am 16.03.2004]«.
- opso3 open source »URL: <http://www.opensource.org> [zuletzt besucht am 16.03.2004]«.
- php-h1 PHP-Homepage.de »URL: <http://www.php-homepage.de/manual/ref.ming.php> [zuletzt besucht am 09.03.2004]«.
- php-h2 PHP-Homepage.de »URL: <http://www.php-homepage.de/manual/functions.swfaction.php> [zuletzt besucht am 15.03.2004]«.
- phpn Php.net »URL: <http://de.php.net/ming> [zuletzt besucht am 25.03.2004]«.
- rask Raskin, Jeff: Das intelligente Interface – Neue Ansätze für die Entwicklung interaktiver Benutzerschnittstellen, Addison-Wesley Longman Verlag, München 2001.
- redd RedDot »URL: : <http://reddot.netmedia.de/flashintegration.php> [zuletzt besucht am 01.03.2004]«.
- roßb Roßbach, Stephan: Der Apache Webserver – Einrichtung und Verwaltung, Addison-Wesley Longman Verlag, München 1999.

- sche Schema »URL <http://www.schema.de/sitehtml/site-d/software0.htm> [zuletzt besucht am 23.03.2004]«.
- sour SourceForge.net »URL: <http://sourceforge.net/projects/ming> [zuletzt besucht am 15.03.2004]«.
- sour2 SourceForge.net »URL: <http://sourceforge.net> [zuletzt besucht am 16.03.2004]«.
- tech TechEncyclopedia »URL: <http://www.techweb.com/encyclopedia> [zuletzt besucht am 15.03.2004]«.
- thro Throll, Markus: MySQL – Grundlagen, Entwicklung, Administration, Galileo Press GmbH, Bonn 2002.
- w3c W3C »URL: <http://www.w3.org/Consortium/Offices/Germany/Trans/WAI/webinhalt.html> [zuletzt besucht am 01.03.2004]«.
- voig Voigt, Ingo/Abel, Holger »URL: [http://www.contentmanager.de/magazin/artikel\\_170\\_web\\_accessibility\\_initiative\\_wai.html](http://www.contentmanager.de/magazin/artikel_170_web_accessibility_initiative_wai.html) [zuletzt besucht am 01.03.2004]«.
- webg1 Webgroove »URL: <http://www.webgroove.de> [zuletzt besucht am 09.03.2004]«.
- webg2 Webgroove »URL [http://www.webgroove.de/webpromo/artist\\_special.pdf](http://www.webgroove.de/webpromo/artist_special.pdf) [zuletzt besucht am 02.03.2004]«.
- ziet Zietz, Christian »URL: [http://www.contentmanager.de/magazin/artikel\\_54\\_da\\_tut\\_sich\\_etwas\\_innerhalb\\_eines\\_cms\\_.html](http://www.contentmanager.de/magazin/artikel_54_da_tut_sich_etwas_innerhalb_eines_cms_.html) [zuletzt besucht am 09.03.2004]«.
- zogg Zogg, Christoph: Die fünf häufigsten CMS-Missverständnisse – Was Sie schon immer über Content-Management-Systeme wissen wollten, ihnen aber niemand zu sagen wagte. In: Futurecom Insights, 2003, H.2, S. 4-5.
- zsch Zschau, Oliver/Traub Dennis/Zahradka Rik: Web Content Management – Websites professionell planen und betreiben, Galileo Press GmbH, Bonn 2002.



## **Anhang**

Dieser Diplomarbeit ist eine CD beigefügt, auf der die Quellcodes und Beispiele zu finden sind, die während den Untersuchungen mit PHP-Ming entstanden sind.

Dies sind eine Internetseite in Flash (hdm.html), die dann in Ming nachprogrammiert wurde (25jam.html) und einige weitere Beispiele.