

SenoCMS, Content Management System für suchmaschinenoptimierten Webseitenaufbau

Systemdokumentation

September 2006
Poltschak Stefanie

Inhaltsverzeichnis

1	Projektbeschreibung	3
2	Systemanforderungen	3
2.1	Webserver	3
2.2	Browser	3
2.3	Einstellungen	3
3	Systemkomponenten	4
3.1	PHP	4
3.2	Smarty Template Engine, SmartyDoc	4
3.3	ADOdb Database Abstraction Library for PHP	4
3.4	Ajax, Xajax	4
3.5	FCKEditor	5
4	Architektur	6
4.1	install	6
4.2	admin	6
4.3	lib	6
4.4	bbclone	6
4.5	includes	6
4.6	mediamanager	7
4.7	templates	7
4.8	templates_c	7
4.9	web	7
5	Umsetzung, CMS Grundfunktionalitäten	7
5.1	Installation SenoCMS	7
5.2	Administrator	7
5.3	Sprachen	7
5.4	Session Management	8
5.5	Nested Set Modell, Abbildung von Baumstrukturen	8
6	Umsetzung, Suchmaschinenoptimierung	9

6.1	URL Rewrite: Apache ForceType Direktive	9
6.2	Struktureller Aufbau, Templategenerierung	10
6.3	Layoutierung mittels CSS	10
6.4	Optimierung durch Tags	10
6.4.1	Title-Tag	10
6.4.2	Meta-Tags	11
6.4.3	Überschriften	11
6.4.4	Alt/Title- Attribute	11
6.5	Contentoptimierung	11
6.5.1	SEO Report	11
6.5.2	Keywords	11
6.5.3	Keyworddichte	12
6.5.4	Position der Keywords im Text	12
6.5.5	Texthervorhebungen	12
6.5.6	Google PageRank	12
6.5.7	Kontrolle der Linkpopularität	13
6.5.8	Eintrag im Verzeichnis DMoz	13
7	Administrationsoberfläche	14
7.1	Icons	14
7.2	SenoCMS Logo	14
7.3	JSCookMenu	14
7.4	OverLIB	14
7.5	Browserkompatibilität	15

1 Projektbeschreibung

SenoCMS ist ein Open Source Web-Content-Management-System das neben Standardfunktionalitäten eines CMS gezielt auf die Optimierung der Seiten für Suchmaschinen ausgelegt ist. Das CMS basiert auf der serverseitigen Skriptsprache PHP, als Datenbank wird MySQL eingesetzt. Der Entwickler wird von Projektbeginn bei der Generierung der „suchmaschinenfreundlichen“ Seiten und deren Verwaltung unterstützt. Hauptpunkte der Optimierung sind ein sauberer struktureller Aufbau, gültiges XHTML, Einsatz von Cascading Stylesheets (CSS) zur Layoutierung um standardkonformen XHTML-Quellcode und zugleich eine optimal für Suchmaschinen aufgebaute Seite garantieren zu können.

Die Optimierung endet nicht mit dem strukturellen Aufbau, sondern setzt sich in der Aufbereitung der Inhalte fort. Auch hier werden Parameter, die das Ranking beeinflussen (wie beispielsweise die Keyworddichte und deren Position im Text) überprüft, und dem Anwender Hinweise sowie Verbesserungsvorschläge geboten. Für jede Seite steht ein Report zur Verfügung der Auskunft über die Keywords und deren Dichte gibt. Als Keyword definierte Wörter werden im Quellcode speziell hervorgehoben um eine höhere Gewichtung zu erreichen. Zur Kontrolle der Linkpopularität zeigt der Report zudem die Anzahl der Backlinks auf die Seite, bezogen aus den Angaben der Suchmaschinen *Google*, *Yahoo!* und *MSN Search* und errechnet für jede Seite den *Google PageRank*.

Zusätzlich baut das CMS auf moderne technische Infrastruktur im Administrationsbereich auf, AJAX (Asynchronous Javascript and XML) kommt zum Einsatz um dem Anwender eine benutzerfreundliche desktopähnliche Web-Oberfläche bieten zu können, in der eine HTML-Seite nicht mit jeder HTTP-Protokoll-Anfrage komplett neu geladen werden muss.

2 Systemanforderungen

2.1 Webserver

SenoCMS setzt Apache als Webserver voraus, um die Technik der *.htaccess*-Dateien und der *Apache ForceType Direktive* einsetzen zu können. Wichtig ist dabei, dass in der *httpd.conf* innerhalb des Programmverzeichnis des Webservers die Nutzung von *.htaccess*-Dateien erlaubt ist.

2.2 Browser

Eine einwandfreie Bedienung der Administrationsfläche kann für Mozilla Firefox (getestet in Version 1.5) und Internet Explorer 6+ garantiert werden.

2.3 Einstellungen

Die Browsereinstellungen müssen Cookies zulassen (zum Zwischenspeichern der Sessiondaten), JavaScript muss aktiviert sein, bei Verwendung von Internet Explorer zusätzlich Active-x.

3 Systemkomponenten

3.1 PHP

SenoCMS basiert auf der serverseitigen Skriptsprache *PHP* und setzt die Installation von *PHP*> Version 4.0.6 auf dem Webserver voraus.

3.2 Smarty Template Engine, SmartyDoc

Als Template Engine ist *Smarty*¹, eine Open Source PHP-Bibliothek, im Einsatz, die bei der Entwicklung von Webapplikationen die Trennung von Code und Ausgabe ermöglicht. Die strikte Trennung von Programmcode (PHP-Code) und Ausgabecode (XHTML-Code) verbessert die Strukturierung der Programme und senkt die Zeit für Layoutanpassungen. Sowohl PHP- als auch XHTML-Code bleiben übersichtlich.

Sowohl für die aus dem CMS dynamisch generierten Seiten, als auch im Administrationsbereich, kommt die Smarty Tempate Engine zum Einsatz. Die Smarty Bibliothek in der Version 2.6.10 ist im Installationspaket enthalten.

3.3 ADOdb Database Abstraction Library for PHP

*ADODB*² ist eine Sammlung von Datenbank-Abstraktions-Funktionen und unterstützt aktuell *MySQL*, *PostgreSQL*, *Oracle*, *Interbase*, *Microsoft SQL Server*, *Access*, *FoxPro*, *Sybase*, *ODBC* und *ADO*. Eine einheitliche und einfache Schnittstelle (API) wird zur Verfügung gestellt, um mit allen unterstützten Datenbanken auf gleiche Art und Weise kommunizieren zu können.

In *ADODB* wird der objektorientierte Ansatz verfolgt, das Verbindungs-Objekt ist verantwortlich für die Verbindung zur Datenbank, für die Formatierung Ihrer SQL-Anweisung und zur Abfrage des Datenbankservers. Das Datensatz-Objekt ist verantwortlich für die Abholung der Daten, sowie zur Formatierung des Ergebnisses als Text oder als Array. Neben der Verbindung zur Datenbank werden etliche nützliche Funktionen zur Standardisierung von SQL-Anweisungen geboten und somit eine einheitliche Schnittstelle für den Zugriff auf die Datenbank geboten. *SenoCMS* setzt standardmäßig *MySQL* als Datenbank ein.

3.4 Ajax, Xajax

Um dem Anwender im Administrationsbereich eine interaktive, Desktop-ähnliche Web-Oberfläche bieten zu können, kommt *AJAX* zum Einsatz. Hierbei handelt es sich um ein Konzept der Datenübertragung zwischen Server und Browser, welches es ermöglicht, dass die HTML-Seite nicht mit jeder HTTP-Protokoll-Anfrage komplett neu geladen werden muss, sondern nur gewisse Teile einer HTML-Seite sukzessive bei Bedarf nachgeladen werden.

Eine Benutzeraktion, die für gewöhnlich eine HTTP-Anfrage erzeugen würde, erzeugt nun einen JavaScript-Aufruf, der an die *AJAX*-Engine delegiert wird. Jede Antwort auf eine Aktion des Nutzers, die keine Verbindung zum Server erfordert, wie beispielsweise das Validieren von Daten, das Verändern von Daten, welche sich im Speicher befinden, und sogar das Navigieren zwischen einzelnen Elementen der Webseite kann von der *AJAX*-Engine bewältigt werden. Benötigt die *AJAX*-Engine

¹<http://smarty.php.net>

²<http://adodb.sourceforge.net>

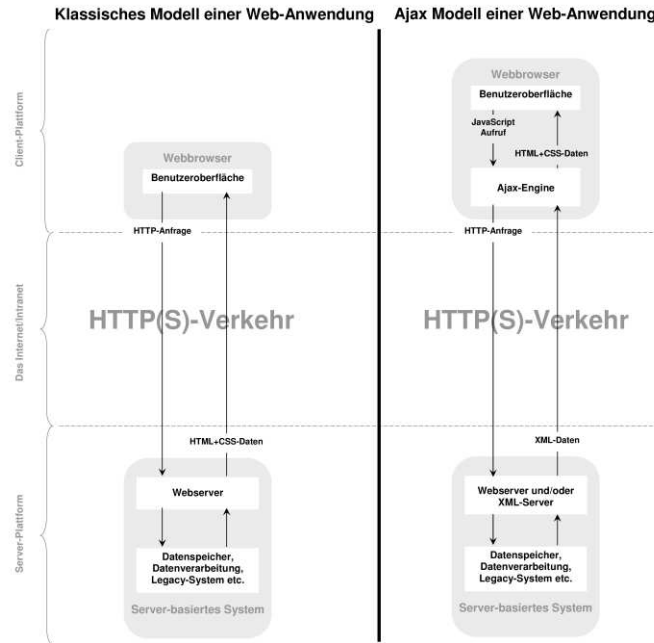


Abb. 3.1: Das Modell einer traditionellen Webanwendung im Vergleich mit einer Ajax Webanwendung.

Daten vom Server, um eine bestimmte Aktion erfolgreich durchführen zu können, es kann sich hierbei beispielsweise um das Übertragen von Daten, die verarbeitet werden müssen, um das Nachladen einzelner Bausteine der Benutzeroberfläche oder um das Laden neuer Daten handeln, führt diese eine asynchrone Anfrage in Form eines XML-Dokuments an den Server durch. Dabei wird jedoch die Interaktion des Benutzers mit der Anwendung, wie dies bei gewöhnlichen Webanwendungen der Fall ist, nicht unterbrochen.

Abbildung 3.1, zeigt das Modell einer traditionellen Webanwendung im Vergleich mit einer AJAX Webanwendung.

SenoCMS setzt *Xajax*³, eine Open Source PHP Klassenbibliothek zur Erzeugung von AJAX Applikationen, die auf (X)HTML, CSS, JavaScript und PHP basieren, ein. *Xajax* ermöglicht den asynchronen Aufruf von serverseitigen PHP Funktionen, indem das *Xajax* PHP Objekt JavaScript Wrapper-Funktionen für die PHP Funktionen generiert, die asynchron von der Applikation aufgerufen werden sollen. Nach dem Aufruf verwenden die Wrapper-Funktionen das JavaScript XMLHttpRequest Objekt, um asynchron mit dem *Xajax* Objekt am Server kommunizieren zu können, das die zugehörige PHP Funktion aufruft. Nach Abhandlung der Funktion wird eine *Xajax* XML-Antwort von der PHP Funktion an die Applikation zurückgegeben. Diese XML-Antwort enthält Anweisungen und Daten, die zur Aktualisierung von Teilen der Applikation verwendet werden.

3.5 FCKEditor

Zur Eingabe und Verwaltung der Inhalte ist der Open Source HTML Texteditor *FCKeditor*⁴ (in der Version 2.2) eingebunden, der vollständige Integration für PHP bietet. Die mittels FCKeditor erstellten Dokumente sind serverseitig XHTML 1.0 kompatibel, die kompakte Oberfläche ist durch das WYSIWYG-Konzept sehr einfach zu handhaben.

³<http://www.xajaxproject.org>

⁴<http://www.fckeditor.net>

Zusätzlich unterstützt der Editor Schrift- und Text-Formatierung, Textfunktionen wie Copy and Paste, Undo, Redo, Erstellen von Links, Setzen von Ankeren, Erstellen und Bearbeiten von Tabellen sowie die Unterstützung von CSS.

Zum Einfügen von Bildern, Flash oder anderen Quellen ist ein Medienmanager inkludiert, der die Dateien am Server strukturiert im Ordner `mediamanager` der Site ablegt. Kleinere Adaptionen des Editors ermöglichen es, dass zwingend ein alternativer Text zu Bildern eingegeben werden muss, ebenso wie das `Title`-Attribut gesetzt werden muss. Wird dieses nicht explizit angegeben, so wird der alternative Bildtext auch für das `Title`-Attribut übernommen. Das Skin des Editors ist dem Erscheinungsbild des CMS angepasst.

4 Architektur

4.1 install

Bei `install` handelt es sich um das Installationsverzeichnis, das zur erstmaligen Installation von *SenoCMS* benötigt wird. Im Laufe des Installationsvorgangs wird die Datenbank eingereichtet sowie der Administrator-Zugang zur Administrationsoberfläche angelegt.

4.2 admin

Admin ist das Hauptverzeichnis von *SenoCMS*, das Konfiguration, inkludierte Klassen, Funktionen, JavaScript, CSS, verwendete Bilder sowie Smarty Templates der Administrationsoberfläche beinhaltet. Für jede Seite im Adminbereich existiert eine PHP-Datei, sowie eine zugehörige Smarty Template Datei, die dafür sorgt den Programmcode in den Ausgabecode einzubinden. Manipulationen an der Datenbank werden über Xajax-PHP-Funktionen realisiert, diese befinden sich in `functions-/xajax` und sind je gleich bezeichnet wie die zugehörige PHP-Datei. Zum Aufbau von verschiedenen Darstellungsarten bspw. von Listen im Administrationsbereich existieren unterschiedliche Funktionen, die mit dem Präfix `draw_` gekennzeichnet sind. Im Ordner `templates_c` sind die kompilierten Templates abgelegt.

4.3 lib

Im Verzeichnis `lib` finden sich die Bibliotheken *adodb*, *dbtree* und *Smarty*, die von *SenoCMS* benötigt werden.

4.4 bbclone

Das Verzeichnis `bbclone` beinhaltet den inkludierten OpenSource Tracker *bbclone*⁵, der im CMS inkludiert ist.

4.5 includes

Im Verzeichnis `includes` befinden sich der *FCKeditor*, sowie Funktionen zum Seiten- und Content-aufbau und zur Menügenerierung.

⁵<http://bbclone.de>

4.6 mediamanager

Die per *FCKeditor* hochgeladenen Bilder, Flash oder sonstige Dateien werden im Verzeichnis *mediamanager* abgelegt, können somit auch direkt in dieses Verzeichnis gespielt werden. Der Ordner *favicon* dient zum Ablegen einer Favicongrafik für die Site. In *thumbs* werden die automatisch generierten Vorschaubilder für die Überblicksdarstellung der Contentelemente gespeichert.

4.7 templates

Im Ordner *templates* findet man die generierten Seitentemplates der Site sowie die zugehörigen CSS-Files, die von hier aus auch vom Entwickler ggf. nachbearbeitet werden können. Die Strukturen zum Aufbau der Seiten, sowie deren CSS-Dateien und Vorschaubildern befinden sich im Ordner *structures*.

4.8 templates_c

Im Ordner *templates_c* sind die kompilierten Templates abgelegt.

4.9 web

Die Datei *web* wird unter Verwendung der *Apache ForceType Direktive* bei jeder Seitenanfrage aufgerufen. Hier werden die Informationen zur angeforderten Seite ermittelt und mit zugehöriger Menüstruktur, Content und Templateinformation die Darstellung der Seite bewerkstelligt.

5 Umsetzung, CMS Grundfunktionalitäten

5.1 Installation SenoCMS

Beim Installationsvorgang wird zunächst die Datenbank eingerichtet. Die Konfigurationsdaten werden in die Datei *admin/includes/configure.php* geschrieben, diese beinhalten, die bei der Installation eingegebenen Verbindungsdaten zur Datenbank, sowie den aus dem aktuellen Verzeichnis ermittelten Pfad des Wurzelverzeichnisses. Änderungen können im Nachhinein in den Konfigurationsdateien vorgenommen werden.

5.2 Administrator

Als zweiter Schritt der Installation steht das Anlegen eines Administrators, der Zugang zur Administrationsoberfläche der Site hat.

5.3 Sprachen

SenoCMS liegt in Version 1.1 in den Sprachen Deutsch und Englisch vor. Die Sprache der Administrationsoberfläche wird durch die Sprachzuweisung beim Benutzer eingestellt, und kann später über die Konfigurationsdatei geändert werden (siehe Benutzerdokumentation). Die Texte sind in

den Konfigurationsdateien `admin/configs/language` (Texte in Smarty-Template-Dateien) und `includes/language` definiert und können hier auch angepasst werden.

5.4 Session Management

Zum Übermitteln von Daten einer Seite an eine andere Seite werden diese in der Session abgespeichert, die beim Login angelegt wird. In der Session sind zudem die Daten des Benutzers und dessen Einstellungen gespeichert, die bei jedem Seitenaufruf überprüft werden. Bei Logout wird die Session wieder zerstört. Cookies müssen im Browser zugelassen werden, um die Sessiondaten zwischenspeichern zu können.

5.5 Nested Set Modell, Abbildung von Baumstrukturen

Um hierarchische Daten in einer Datenbank speichern zu können, stellt sich die Frage nach einem Modell zur Abbildung von Baumstrukturen. Benötigt werden solche Abbildungen in einem CMS bspw. beim Aufbau des Seitenbaums, wenn Seiten in mehreren Ebenen angeordnet oder später auch neu sortiert werden sollen. Nach einiger Recherche und dem Vergleich von unterschiedlichen Modellen stellte sich für diese Anwendung das „Nested Set Modell“ als brauchbarer Ansatz heraus.

Die Idee, welche sich hinter diesem Begriff *Nested Sets* verbirgt, ist die Abstraktion von Bäumen als Mengen und Teilmengen oder anders gesagt: verschachtelte Mengen.

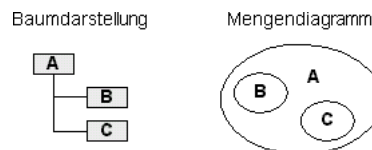


Abb. 5.1: Baum und Menge

Wie in Abbildung 5.1 sehr gut zu erkennen ist, lassen sich baumartige Strukturen leicht als Mengen und Teilmengen darstellen. Die Wurzel A enthält die Mengen der Objekte B und C. Die Abbildung lässt sich als Tabelle darstellen, wie Abbildung 5.2 zeigt.

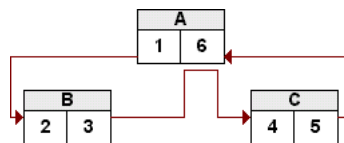


Abb. 5.2: Baumdarstellung in Tabellen

Der Aufbau zeigt drei Teile: den Kopf des Knotens, sowie eine linke (LFT) und eine rechte Seite (RGT), die dafür benötigt werden, um die Abhängigkeiten innerhalb des Baumes darstellen zu können. Die Reihenfolge der Knoten wird durch das Auslesen der Zahlen, mittels so genanntem Preorder-Walk, gewährleistet. Bei den einzelnen Elementen des Baumes wird die linke Seite des Wurzelknotens ausgelesen und dann alle linken Seiten der Unterknoten durchlaufen bis zum letzten Blatt, dann werden alle rechten Seiten ausgelesen. Die Darstellung mit den Tabellen zeigt deutlich die Logik hinter den Nested Sets.

Die Wurzel beginnt links immer mit 1. Danach werden in numerischer Reihenfolge zuerst alle linken

und dann alle rechten Seiten durchlaufen. Abbildung 5.3 stellt die Baum- neben die Tabellendarstellung, um zu zeigen, wie diese beiden Darstellungen einander entsprechen.

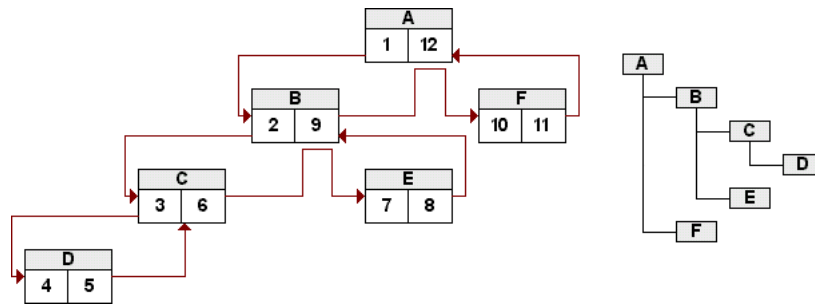


Abb. 5.3: Umfangreicherer Baum als Tabellen- und Baumdarstellung

Der Vorteil dieses Modells ist, dass es Regeln gibt, die für diese Bäume gelten und über die Manipulationen in der Baumdarstellung (Einfügen, Löschen neuer Zweige und Blätter etc.) durch relativ einfache Änderungen in der Datenbank durchgeführt werden können. Zu diesen Regeln zählen:

- $LFT = 1 \Rightarrow$ eine Wurzel hat auf der linken Seite immer eine 1 stehen, dies ist der Wurzelknoten.
- Blatt $RGT - Blatt LFT = 1$
- $Wurzel RGT / 2 =$ Anzahl der Knoten im Baum \Rightarrow teilt man in der Wurzel den Wert der rechten Seite, so erhält man die Anzahl aller Knoten im Baum
- $\text{floor}((RGT - LFT) / 2) =$ Anzahl der Kindknoten im Zweig (incl. der Blätter) \Rightarrow zieht man von der rechten die linke Seite ab und teilt sie durch 2, so entspricht das gerundete Ergebnis der Anzahl der Blätter im Zweig
- alle LFT- und RGT-Werte sind eindeutig!

Die PHP Bibliothek *dbtree*⁶ implementiert den Ansatz der Nested Sets zum Speichern von baumähnlichen Strukturen in einer Datenbank und wird von *SenoCMS* eingesetzt.

6 Umsetzung, Suchmaschinenoptimierung

6.1 URL Rewrite: Apache ForceType Direktive

Die von *SenoCMS* generierten Seiten werden dynamisch mit Inhalten aus der Datenbank generiert, was für das Aussehen der URL prinzipiell bedeuten würde, dass ein Parameter darüber Auskunft geben muss welche Inhalte dargestellt werden sollen. Häufig sind solche von Content Management Systemen dynamisch generierten Seiten durch den Aufbau der URL in einer Form wie `http://www.mysite.com/index.php?id=28` erkennbar. Eine solche URL bringt zwei Nachteile mit sich, erstens ist es für den Benutzer alleine über die URL schwer erkennbar in welchem Bereich der Seite er sich gerade befindet, zweitens werden Seiten deren Adresse (URL) nach „Dynamik“ aussieht von vielen Suchmaschinen prinzipiell nicht indiziert. Der Grund dafür liegt darin, dass es für Suchmaschinen wenig Sinn macht Inhalte zu indizieren bei denen zu einem späteren Zeitpunkt

⁶<http://dev.e-taller.net/dbtree>

der Inhalt der Seite nicht mehr nachvollziehbar ist, wie es bspw. der Fall sein könnte wenn in einer dynamischen URL die SessionID mitübergeben wird.

Natürlich möchte man deswegen nicht darauf verzichten müssen von Suchmaschinen indiziert zu werden. Aus diesem Grund gibt es eine Reihe von „Tricks“ wie dynamische Seiten als statisch simuliert werden können.

Eine elegante Methode zum URL Rewrite bietet Apache mit der ForceType Direktive⁷, welche von SenoCMS verwendet wird. Apache kann einer Datei explizit einen MIME-Typ zuweisen. Einem PHP-Script bspw. „web“ (ohne Endung) wird mit Apaches Force-Type-Direktive der Typ application/x-httpd-php zugewiesen (per htaccess Datei).

Die URL lautet nun bspw. `http://www.mysite.com/web/Information/`. Der Name des URL-Präfixes (hier bspw. „web“) ist mit „web“ vordefiniert und kann über die Konfigurationsdatei auch im Nachhinein geändert werden (siehe Benutzerdokumentation). Die URL ruft nun web als PHP-Script auf, hier wird der Pfad analysiert und daraus über die Datenbank eindeutig festgestellt welche Seite und somit welche Inhalte dargestellt werden sollen. Lediglich innerhalb eines Navigationspunkt auf einer Ebene liegende Seiten mit ident gleichem Namen könnten damit nicht unterschieden werden.

Leerzeichen zwischen Wörtern eines Seitennames werden in Bindestriche umgesetzt, deutsche Umlaute wie bspw. ä werden aufgelöst zu bspw. ae. Alle Suchmaschinen indizieren eine solche Seite problemlos, weil kein Unterschied zu statischen HTML-Seiten mehr vorhanden ist.

6.2 Struktureller Aufbau, Templategenerierung

Bevor ein Template als Designvorlage einer Seite generiert werden kann muss der Entwickler die Strukturtemplates sowie die zugehörigen CSS-Dateien zur Layoutierung bereit gestellt haben. Diese müssen im Order „templates/structures“ und die CSS-Datei/en im weiteren Unterordner „css“ abgelegt sein.

Beim Anlegen eines Templates über die Administrationsfläche werden die Strukturdateien als Vorlage genommen, die Sektionen entsprechend den Eingaben des Benutzers benannt und eine Template-Datei sowie die zugehörige CSS-Datei geschrieben.

Zuständig für die Erzeugung der Dateien sind ist „admin/generate_templatecss.php“ sowie die darin aufgerufenen Funktionen aus „admin/includes/functions/generate.php“.

6.3 Layoutierung mittels CSS

Die Layoutierung der Sites erfolgt, sofern der Entwickler bei der Erstellung der Grundstruktur/en die Vorgaben eingehalten hat, ausschließlich per CSS. Zusätzlich zu den CSS-Dateien der Templates können CSS-Dateien angelegt werden die mehreren Templates zugehörig sind.

6.4 Optimierung durch Tags

6.4.1 Title-Tag

Der Title-Tag kann für jede Seite gesondert angegeben werden. Wird das Eingabefeld leer gelassen so wird standardmäßig eine Title-Information aus Seitenname und Bezeichnung der Site generiert,

⁷<http://www.devarticles.com/c/a/Apache/Using-ForceType-For-Nicer-Page-URLs/>

der natürlich jederzeit beliebig umbenannt werden kann.

6.4.2 Meta-Tags

Ebenso können auch die Meta-Informationen keywords und description für jede Seite separat angegeben werden. Möchte man eine Standarddescription für die ganze Site angeben (was zwar wenig Sinn macht, wegen der Nichtbeachtung der Meta-Tags von vielen Suchmaschinen aber durchaus praktiziert werden kann), so kann man diese globalen Metadaten in der Konfiguration der Site definieren. Der Meta-Tag „language“ wird mit der bei der Site eingestellten Sprache gesetzt. Die Meta-Informationen werden bei jedem Template eingebunden, dies geschieht über das Template „header.tpl“ das in jedem Template inkludiert wird.

6.4.3 Überschriften

Zu jedem Contentelement kann eine Überschrift eingetragen werden, die auf den Seiten innerhalb von „H-Tags“ ausgegeben wird, „H1“-“H6“ kann für jede Überschrift gesondert eingestellt werden. Die Formatierung der „H-Tags“ kann per CSS erfolgen.

6.4.4 Alt/Title- Attribute

Durch das Setzen der Alt- und Title-Attribute bei Bildern können auf einer Seite zusätzliche themenrelevante Informationen und Schlüsselwörter untergebracht werden. Um diese beiden Attribute nicht leer lassen zu können ist das Eingabefeld für alternativen Text bei Bildern über den FCKE-Ditor als Pflichtfeld definiert, wird das Title-Attribut nicht angegeben so wird der ALT-Text auch als Title übernommen. Das Title-Attribut sollte außerdem bei Links gesetzt werden. Bei den Seiten steht ein Eingabefeld für das Link Title-Attribut (als beschreibenden Linktext) zur Verfügung, wird dieses leer gelassen so wird als Title-Attribut der Name der Seite übernommen.

6.5 Contentoptimierung

6.5.1 SEO Report

Zur Überwachung der Optimierungsmöglichkeiten des Contents wird für jede Seite ein Report generiert der über die definierten Schlüsselwörter der Seite und deren Vorkommen im Content Auskunft gibt. Für die Generierung des Reports zeigt sich die Funktion „admin/includes/functions/xajax/kw_density_analysis.php“ verantwortlich.

6.5.2 Keywords

Die Schlüsselwörter können entweder global oder seitenbezogen über den Meta-Tag „keywords“ festgesetzt, oder direkt über den SEO-Report als Keywords definiert werden. Letzteres funktioniert per Klick auf das Schlüsselsymbol neben einem der vorgeschlagenen Wörter. Als potenzielle Schlüsselwörter werden jene Wörter angesehen, die ein gewissen Mindestvorkommen im Text aufweisen (die Parameter lassen sich hier frei konfigurieren) und nicht als Stoppwörter definiert wurden. Die Dateien mit den Stoppwörtern existieren für jede Sprache im Verzeichnis „admin/includes/language“ und können dort auch ergänzt oder abgeändert werden.

6.5.3 Keyworddichte

Für jedes potenzielle Schlüsselwort wird zusätzlich die Dichte bezogen auf alle Wörter im Content errechnet und angegeben. Ein Kennsymbol gibt für definierte Schlüsselwörter darüber Auskunft ob die Dichte innerhalb der als optimaler Keyword-Dichte definierten Schranken liegt.

6.5.4 Position der Keywords im Text

Da auch das Vorkommen der Schlüsselwörter bezogen auf die Position im Text eine bedeutende Rolle spielt, wird außerdem kontrolliert ob die Schlüsselwörter auch in den ersten bspw. 1000 Zeichen (dieser Wert kann über die Konfiguration der Site angepasst werden) öfter als zweimal vorkommt. Ist dies nicht der Fall, so ist dies im Report besonders gekennzeichnet.

6.5.5 Texthervorhebungen

Die als Schlüsselwörter einer Seite definierten Wörter werden bei Anzeige der Seite im Quellcode mittels dem „strong-Tag“ speziell hervorgehoben um deren Gewichtung zu heben. Die Darstellung des „strong-Tags“ ist per CSS so definiert, dass nicht jedes der Schlüsselwörter fett gedruckt erscheint.

6.5.6 Google PageRank

Der SEO-Report ermittelt zusätzlich für jede Seite gesondert den *Google PageRank* und zeigt diesen grafisch ausgedrückt auf einer Skala von 0 bis 10 an, wie auch in Abbildung 6.1 zu sehen.

Zur Ermittlung des *Google PageRank* ist es entscheidend zu wissen, dass der Algorithmus nicht öffentlich verfügbar ist und der PageRank auch über die *Google-API* nicht abgerufen werden kann. Bei genauerer Inspektion der *Google-Toolbar*, die den PageRank darstellt, stellt sich heraus, dass über die aktuelle URL eine Checksumme gebildet wird, mit deren Wissen der Wert des PageRanks über einen Aufruf der URL: `http://www.google.com/search?client=navclient-auto&ch=[checksum]&features=Rank&q=info:[url]` zurückgegeben wird. Dies hat einige Programmierer dazu motiviert, den in Form von DLLs in der *Google-Toolbar* inkludierten Programmcode zu disassemblieren, also den dahinter liegenden Algorithmus zu knacken und eigene Generatoren für die Checksummen-Berechnung zu entwickeln. Im Juni 2004 wurde der Checksummen-Algorithmus erstmals gecrackt. Der Code verbreitete sich schnell übers Web und führte zur verbreiteten Entwicklung von Tools zur Anzeige des PageRanks, abgesehen von der *Google-Toolbar*. Die öffentliche Verbreitung des Source-Codes liegt natürlich nicht im Interesse des Suchmaschinenbetreibers, weshalb *Google* auch immer wieder mit Änderungen im Checksummen-Algorithmus reagiert.

Die von *SenoCMS* eingesetzte Klasse bedient sich zur Ermittlung des *Google PageRanks* des im Internet leicht zu findenden Checksummen-Algorithmus. Um dessen Verwendung gegenüber *Google* nicht zu offensichtlich zu gestalten, wird für jede Abfrage ein aus einer Liste zufällig ermittelter *Google-Hostname* ausgewählt.

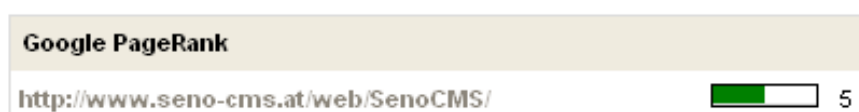


Abb. 6.1: Anzeige des Google PageRank im SEO-Report.

6.5.7 Kontrolle der Linkpopularität

Zur Kontrolle der Linkpopularität einer Seite bietet der SEO-Report eine Übersicht über die eingehenden Links. Die resultierend aus einer Backlink Analyse bei den Suchmaschinen *Google*, *Yahoo!* und *MSN Search* ermittelte Anzahl an Backlinks wird angezeigt, verlinkt auf die Ergebnisseiten der Analyse für eine detailreichere Auflistung. Zu beachten ist, dass die Anzahl der ermittelten Backlinks, hinsichtlich der Vollständigkeit nicht als zuverlässig angesehen werden können. Die Ergebnisse können dennoch zur ungefähren Einschätzung verwendet werden.

Die Backlink-Abfrage bei *Google*, *Yahoo* und *MSN Search* passiert über die drei URLs:

[http://www.google.com/search?hl=en&lr=&ie=UTF-8&q=link:\[URL\]](http://www.google.com/search?hl=en&lr=&ie=UTF-8&q=link:[URL])

[http://search.yahoo.com/search?p=link:\[URL\]](http://search.yahoo.com/search?p=link:[URL])

[http://search.ninemsn.com.au/results.aspx?q=link:\[URL\]](http://search.ninemsn.com.au/results.aspx?q=link:[URL])

Aus der textuellen Angabe auf den Ergebnisseiten der Suchanfragen wird dann die Anzahl an Backlinks extrahiert. Die drei Suchmaschinen zeigen eine unterschiedliche Information an, weshalb die Extraktion der Anzahl an Ergebnissen dem Muster der jeweiligen Suchmaschine anzupassen ist. Im Falle einer Änderung des angezeigten Textes, muss diese Anpassung auch im Programmcode vorgenommen werden. Momentan werden die Zahlen aus folgender, im Header der Suchmaschinen-Ergebnisseiten, präsentierter Information extrahiert:

Google: Results 1 - 10 of about **182** linking to www.seno-cms.at.

Yahoo!: 1 - 10 of about **2,050** for link:<http://www.seno-cms.at>

MSN Search: Page 1 of **8,113** results containing link:www.seno-cms.at

Link Popularität	
Suchmaschinen	Indizierte Links
Google	182
Yahoo!	2050
MSI Search	8113

Abb. 6.2: Übersicht über eingehende Links gefunden von Google, Yahoo! und MSN Search.

6.5.8 Eintrag im Verzeichnis DMoz

Ein Eintrag im Open-Directory Webkatalog DMoz kann zu einem bedeutenden Rankingschub verhelfen. Der SEO-Report überprüft deshalb für jede Seite, ob ein Eintrag im DMoz Verzeichnis existiert und zeigt das Ergebnis an, wie in Abbildung 6.3 zu sehen.

Auf ähnliche Weise wie die Ermittlung der Anzahl an Backlinks wird auch das Bestehen eines Eintrags im DMoz Verzeichnis überprüft. Der Aufruf der URL: [http://search.dmoz.org/cgi-bin/search?search=\[URL\]](http://search.dmoz.org/cgi-bin/search?search=[URL]) liefert eine Seite zurück, die den Text „No Open Directory Project results found“ enthält, so die Seite nicht im Verzeichnis aufgenommen ist. Anderenfalls wird diese Information nicht ausgegeben, was gleichzeitig impliziert, dass ein Eintrag gefunden wurde und in der deutschen Version von *SenoCMS* die Information „Gefunden“ ausgibt.

Eintragungen in Verzeichnissen	
Verzeichnis	Ergebnis
Dmoz	Gefunden

Abb. 6.3: Überprüfung ob ein Eintrag der Seite im DMoz Verzeichnis existiert.

7 Administrationsoberfläche

7.1 Icons

Die verwendeten Icons in der Administrationsoberfläche stammen größtenteils aus der Open Source Icon-Bibliothek von Gnome⁸. Den Anforderungen gemäß wurden die Symbole adaptiert bzw. eigene Icons gestaltet.

7.2 SenoCMS Logo

SenoCMS steht für *Search ENgine Optimized Content Management System*, das Logo (siehe Abbildung 7.1) zeigt einen weißen Hut auf dem S der Schriftzugs. Dies soll, bezogen auf den Begriff *White Hat Search Engine Optimization* (deutsch: ethische Suchmaschinen-Optimierung), auf den Verzicht von verbotenen Praktiken zur Suchmaschinen-Optimierung hinweisen. Für die Gestaltung des Logos geht ein herzlicher Dank an Dipl.-Ing.(FH) Daniela Würthinger.



Abb. 7.1: SenoCMS Logo

7.3 JSCookMenu

Das Hauptmenü der Administrationsoberfläche bedient sich des *JSCookMenus*⁹ (in der Version 1.4.4), einem leistungsfähigen JavaScript Menüscript. Das Script ist in der Lage, auch komplexe Menüs umzusetzen und wird so in einigen bekannten GUI Applikationen eingesetzt.

7.4 OverLIB

*OverLIB*¹⁰ ist eine JavaScript Bibliothek, mit deren Unterstützung die kleinen Tooltipp Popup-Informationsboxen realisiert wurden. Der Benutzer erhält bei rollover über die Informationssymbole zu diesem Thema relevante Tipps, ebenso wie bei den Werkzeugen in der Toolbar. Auf der Contentüberblickseite kommt die Bibliothek zusätzlich zum Einsatz, um im Content enthaltene Bilder bei rollover über den Dateinamen in Vorschau zu zeigen.

⁸<http://art.gnome.org/themes/icon>

⁹<http://jscook.sourceforge.net/JSCookMenu/>

¹⁰<http://www.bosrup.com/web/overlib>

7.5 Browserkompatibilität

Eine einwandfreie Bedienung der Administrationsoberfläche kann für Mozilla Firefox 1.5 und Internet Explorer 6 garantiert werden. Um auch im Internet Explorer (in Versionen vor 7) Bilder im PNG-Format mit transparentem Hintergrund korrekt anzeigen zu können, sorgt eine JavaScript-Funktion¹¹ dafür, automatisch alle auf der Seite befindlichen PNG's mit einem Alpha-Layer zu versehen, mit dem der Internet Explorer umgehen kann.

¹¹<http://homepage.ntlworld.com/bobosola/pngtestfixed.htm>