

DIPLOMARBEIT

Sucherweiterung für das Content Management System TYPO3

Mathias Moser

tm0310038044

Ausgeführt zum Zweck der Erlangung des akademischen Grades

Dipl.-Ing. (FH) Telekommunikation und Medien

am Fachhochschul-Diplomstudiengang

Telekommunikation und Medien St. Pölten

Erstbetreuer: Dipl. Ing. (FH) Fritz Grabo

Zweitbegutachter: Dipl. Ing. Grischa Schmiedl

St. Martin, am 5. September 2007

Inhaltsverzeichnis

Erklärung	5
Kurzfassung	6
Abstract	7
1 Einleitung	8
2 TYPO3 Grundlagen	9
2.1 Geschichte	9
2.2 Lizenzierung	10
2.3 Architektur	11
2.3.1 Systemaufbau	11
2.3.2 Datenbank.....	12
2.3.2.1 Allgemein	12
2.3.2.2 Wichtige Tabellen in der TYPO3-Datenbank	12
2.3.3 Verzeichnisstruktur.....	13
2.4 Das Backend	13
2.5 TypoScript	15
2.6 Designvorlagen	16
2.7 Caching	17
2.8 Community	18
2.9 Weitere Features	19
3 TYPO3 Extension-Programmierung	21
3.1 Grundlagen	21
3.1.1 Namenskonventionen	21
3.1.2 Entwicklungsstatus einer Erweiterung.....	22
3.1.3 Einteilung von Erweiterungen.....	23
3.1.4 Dateistruktur einer Erweiterung.....	23
3.1.5 Der Extension-Manager	25
3.1.6 Online-Extension-Repository	25
3.1.7 Das Modul „Extension Manager“	26
3.2 Backend-Programmierung	27
3.2.1 Eingabemasken.....	27
3.2.2 Das Table Configuration Array „\$TCA“	27

3.2.3	Module.....	28
3.3	Frontend-Programmierung.....	29
3.3.1	Das Frontend-Rendering.....	29
3.3.1.1	„\$TSFE“ (TypoScript-Frontend)	29
3.3.1.2	Der Rendering-Prozess	30
3.3.2	Plugins.....	30
3.4	Flexforms.....	32
3.5	Mehrsprachigkeit.....	32
3.6	Extension „Kickstarter“.....	33
4	Erweiterung planen.....	34
4.1	Anforderungsdefinition.....	34
4.1.1	Was soll durchsucht werden?	34
4.1.2	Reihung der Ergebnisse.....	34
4.1.3	Spracheinstellungen.....	35
4.1.4	Templates	35
4.1.5	Statistik.....	35
4.1.6	Usability.....	35
4.1.7	Dokumentation.....	36
4.2	Analyse bestehender Sucherweiterungen.....	36
4.2.1	Erweiterung „Indexed Search Engine“	37
4.2.1.1	Architektur	37
4.2.1.2	Technische Details	38
4.2.1.3	Datenbanktabellen	39
4.2.1.4	Reihung der Suchergebnisse	39
4.2.2	Erweiterung „Content Search“	39
4.2.2.1	Architektur	40
4.2.2.2	Technische Details	40
4.2.2.3	Datenbanktabellen	40
4.2.2.4	Reihung der Suchergebnisse	40
4.2.3	Erweiterung „[AN] Search It“	41
4.2.3.1	Architektur	41
4.2.3.2	Technische Details	41
4.2.3.3	Datenbanktabellen	42
4.2.3.4	Reihung der Suchergebnisse	42
4.2.4	Erweiterung „AJAX Searching Engine“	42
4.2.4.1	Architektur	42
4.2.4.2	Technische Details	43
4.2.4.3	Datenbanktabellen	43
4.2.4.4	Reihung der Suchergebnisse	43
4.3	Zusammenfassung.....	44
4.4	Schlussfolgerung.....	45
5	„Intelligent Search Engine“.....	46
5.1	Grundlegendes.....	46
5.1.1	Entwicklungsumgebung	46
5.1.2	Extension-Key.....	46

5.1.3	Entwicklungsstatus	47
5.1.4	Verwendung der Extension „XAJAX“	47
5.2	Architektur	47
5.2.1	Systementwurf.....	48
5.2.2	Frontend-Plugin.....	48
5.2.3	Backend-Modul	48
5.2.4	Datenbank.....	49
5.2.4.1	Tabelle „tx_isearch_swords“	49
5.2.4.2	Tabelle „tx_isearch_results“	49
5.3	Umsetzung.....	50
5.3.1	Was soll durchsucht werden?	50
5.3.2	Reihung der Ergebnisse	51
5.3.2.1	Bewertungen.....	51
5.3.2.2	Anzahl der Klicks	52
5.3.2.3	Anzahl der Treffer in einem Ergebnis	52
5.3.2.4	Startpunkte	53
5.3.2.5	Beispiel für eine Gesamtpunkteberechnung	53
5.3.3	Spracheinstellungen	54
5.3.4	Templates	54
5.3.5	Statistik.....	55
5.3.6	Usability.....	56
5.3.7	Dokumentation.....	58
5.4	Implementierung	59
5.4.1	Start mit Extension „Kickstarter“	59
5.4.2	Das Frontend-Plugin	61
5.4.2.1	Die Konfiguration per Flexform.....	61
5.4.2.2	Konfiguration per TypoScript.....	63
5.4.2.3	Die Datei „class.tx_isearch_pi1.php“	63
5.4.2.4	Der Suchvorgang.....	66
5.4.2.5	Die Autovervollständigung.....	68
5.4.2.6	Der Bewertungsvorgang.....	69
5.4.2.7	Vorgang bei Klick auf ein Resultat.....	70
5.4.2.8	Der Reihungsvorgang	70
5.4.3	Das Backend-Modul	71
5.4.3.1	Die Datei „index.php“	72
5.4.3.2	Flexform- bzw. TypoScript-Konfiguration	74
5.5	Veröffentlichung der Extension.....	76
6	Fazit und Ausblick	78
	Literaturverzeichnis	80
	Abbildungsverzeichnis.....	81
	Tabellenverzeichnis	82
	Listings	83
	Inhalt der beiliegenden CD.....	84

Erklärung

Ich versichere, dass

- ich diese Diplomarbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Diplomarbeitsthema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der von den Begutachtern beurteilten Arbeit überein.

Ort, Datum

Unterschrift

Kurzfassung

TYPO3 ist ein modulares auf der Skriptsprache PHP basierendes Open-Source Content Management System für mittlere bis große Websites. Der modulare Aufbau ermöglicht eine leichte Funktionserweiterung durch so genannte Extensions.

Es gibt eine Vielzahl von Erweiterungen für TYPO3, die frei zum Download zur Verfügung stehen. Darunter befindet sich auch eine Reihe von Sucherweiterungen. Viele dieser Extensions liefern jedoch oft nicht die gewünschten Suchergebnisse. Die gefundenen Datensätze beinhalten zwar die gesuchten Wörter, haben aber thematisch nur am Rande damit zu tun. Um ein besseres Suchergebnis zu erhalten, soll der Benutzerin/dem Benutzer daher die Möglichkeit gegeben werden, Suchtreffer nach deren Relevanz zu bewerten. Der Suchalgorithmus soll diese Information bei der Suche berücksichtigen und somit von der Benutzerin/vom Benutzer lernen.

Diese Arbeit beschäftigt sich mit der Lösung und Umsetzung dieser Anforderung. Zunächst wird der Leserin/dem Leser ein Überblick über das Content Management System TYPO3 vermittelt. Detaillierter wird auf die Extension-Programmierung unter TYPO3 eingegangen. Nach den Grundlagen werden die Anforderungen an die zu entwickelnde Sucherweiterung definiert und bestehende Sucherweiterungen auf Grund dieser Anforderungen verglichen. Es wird entschieden, ob Synergien bzw. Vorgehensweisen anderer Erweiterungen genutzt werden können. Abschließend wird auf die Umsetzung der Extension eingegangen und wichtige Entscheidungen betreffend der Implementierung angesprochen.

Abstract

TYPO3 is a modular open-source Content Management Framework built with PHP. The modular structure results in an easily adding and modifying behaviour of the system with so called extensions.

TYPO3 offers many free available extensions for any behaviour. Under these extensions there are also some search extensions. But often there is the problem, that they do not deliver the appreciated results. The results may contain the searched word, but do not fit thematically. To get better results, the user should be integrated. She/He should have the possibility to evaluate each result concerning its relevance. The search algorithm should consider these evaluations to learn from the user.

This thesis has the issue of finding a solution for this requirement. At the beginning, the reader gets an overview of the Management Framework TYPO3, followed by a detailed look at the extension-programming under TYPO3. After the fundamentals, the requirements for the new extension are defined. Furthermore, existing search extensions are compared considering the defined requirements. Afterwards, there is the decision whether there are synergies or procedural methods which can be used. A final look will be taken at the realization, as well as the important decisions, concerning the implementation of the extension.

1 Einleitung

Zielsetzung und Aufbau der Arbeit

Welche detaillierten Anforderungen soll die Sucherweiterung erfüllen? Können Synergien bzw. Vorgehensweisen anderer Sucherweiterungen genutzt werden? Wie findet die Gewichtung und somit die Reihung der Suchtreffer statt? Wie ist die Erweiterung technisch realisierbar?

Im Laufe dieser Arbeit werden die genannten Fragen angesprochen und erläutert. Es wird eine konkrete Lösung gefunden und in Form einer TYPO3-Extension umgesetzt. Zusätzlich wird die fertige Erweiterung der TYPO3-Community unter der GPL (GNU General Public License) zur Verfügung gestellt.

Kapitel 2: „TYPO3 Grundlagen“ verschafft der Leserin/dem Leser einen groben Überblick über das Content Management System TYPO3.

Kapitel 3: In „**TYP03 Extension-Programmierung**“ wird detaillierter auf die Programmierung einer TYPO3-Erweiterung eingegangen.

Kapitel 4: „Erweiterung planen“ beschäftigt sich mit der Definition der Anforderungen, die die fertige Erweiterung erfüllen soll. Außerdem werden 4 Sucherweiterungen analysiert. Abschließend wird entschieden, ob Synergien bzw. Vorgehensweisen der untersuchten Erweiterungen genutzt werden können.

Kapitel 5: „Intelligent Search Engine“ beschäftigt sich mit der Umsetzung und Implementierung der Sucherweiterung. Zusätzlich wird kurz auf die Veröffentlichung eingegangen.

Kapitel 6: „Fazit und Ausblick“ bilanziert die erstellte Erweiterung und versucht einen kleinen Blick in die Zukunft der Extension zu werfen.

2 TYPO3 Grundlagen

TYPO3 ist ein komplexes Content Management System. Um einen Überblick über die Funktionalität von TYPO3 zu schaffen, beschäftigt sich dieses Kapitel mit den Grundlagen des Systems. Es werden grundlegende Themen behandelt, um den Einstieg in die Welt von TYPO3 zu erleichtern.

2.1 Geschichte

„TYPO3 ist ein vom Kopenhagener Programmierer Kasper Skårhøj seit 1997 entwickeltes Web-Content-Management-System (WCMS), das sowohl der Entwicklung als auch der Pflege einer Webseite dient.“ ([1], S. 26)



Abbildung 1: Offizielles TYPO3-Logo

Zu Beginn setzte man auf eine kommerzielle Entwicklung. Dazu wurde TYPO3 1998 der Werbeagentur Superfish.com anvertraut. Ende 1998 wurde das CMS als Unternehmenslösung für Content Management auf der IFRA Ausstellung in Lyon (Frankreich) vorgestellt. Erste Erfahrungen mit diversen Kundinnen/Kunden erforderten 1999 Änderungen im Kern des Systems. Es wurde daher ein neuer, beständiger Kern von Grund auf entwickelt. Dieser ist noch heute die Grundlage für das System. Im Sommer 1999 trennte sich Kasper Skårhøj von Superfish.com. Er entschloss sich, TYPO3 in Eigenregie weiter zu entwickeln. [7]

Seit dem Jahre 2000 ist TYPO3 als Open-Source-Software unter der GPL (GNU General Public License) freigegeben. Betreut wird es seit 2002 von einer stetig wachsenden Community. (vgl. [1], S.26)

Das erste „echte“ TYPO3-Release wurde am 24. Mai 2002 mit der Version 3.0 veröffentlicht - ein Meilenstein in der Geschichte von TYPO3. [7]

Eine der bedeutendsten Neuerungen wurde mit der Version 3.5.1 veröffentlicht, der „Extension-Manager“ (siehe auch Kapitel 3.1.5). Dieser macht eine modulare Erweiterbarkeit des Systems möglich, und schafft damit die Möglichkeit, eigenständig geschriebene Erweiterungen - in TYPO3-Kreisen „Extensions“ genannt - leicht in TYPO3 einzubinden.

2.2 Lizenzierung

TYPO3 ist als Open-Source-Software (GPL – GNU General Public License) freigegeben. Eine Kopie der GPL finden Sie unter <http://www.opensource.org/licenses/gpl-licenses.php>.

Die GPL besagt, dass Software, die unter der GPL besteht

- beliebig zu nutzen ist.
- beliebig vervielfältigt, weiterverarbeitet und verändert werden darf.
- bei Änderungen wieder unter die GPL zu stellen ist.

TYPO3 greift standardmäßig auf die freie Software PHP, MySQL und den Webserver Apache zurück. Es besteht aber auch die Möglichkeit, einen alternativen Webserver oder eine alternative Datenbank zu verwenden.

(vgl. [2], S. 26)

2.3 Architektur

2.3.1 Systemaufbau

TYPO3 gliedert sich in den Systemkern (CORE) und in Extensions, die die Funktionalität des Systems erweitern.

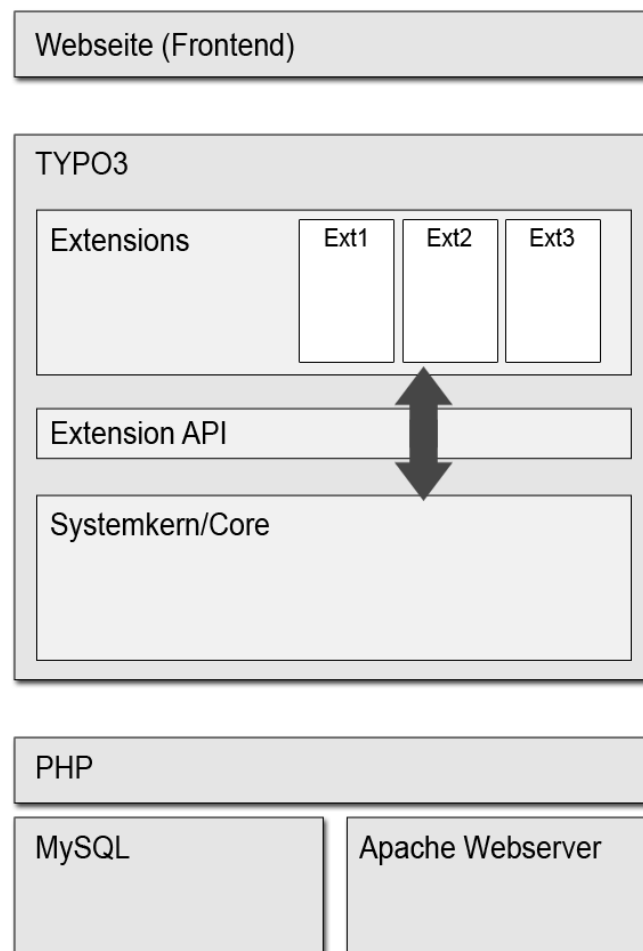


Abbildung 2: Systemaufbau von TYPO3
(vgl. [2], S.31)

Der TYPO3-Systemkern beinhaltet nur die essentiellen Funktionen zur Datenbank-, Datei- und Benutzerverwaltung und bildet das Grundgerüst des Systems. Alle zusätzlichen Funktionen werden durch Extensions realisiert. Sie greifen

über eine Schnittstelle (Extension-API) auf den Systemkern zu. Dies führt zu einer hohen Flexibilität, da somit eine leichte Funktionserweiterung bzw. Funktionsadaption ermöglicht wird. (vgl. [2], S. 32f)

„Die Ausbaufähigkeit durch Extensions ist somit ein zentrales Feature von TYPO3, das es von vielen anderen Systemen abhebt.“ ([2], S. 33)

Es ist auch möglich, mit nur einem Systemkern mehrere TYPO3-Installationen zu betreiben. Diese Installationen können Erweiterungen gemeinsam nutzen („Globale Extensions“) oder individuell eigene Erweiterungen verwenden („Lokale Extensions“). (vgl. [2], S. 32)

2.3.2 Datenbank

2.3.2.1 Allgemein

Standarddatenbank für jede TYPO3-Installation ist die relationale Open-Source-Datenbank MySQL. Seit der Version 3.6 ist TYPO3 mit einem Datenbankabstraktionslayer (DBAL) ausgestattet, der prinzipiell auch die Verwendung von alternativen Datenbanken wie PostgreSQL, Oracle, Firebird, usw. erlaubt. Seit der Version TYPO3 4.0 ist das Datenbank-Abstraktionsmodell offiziell integriert. (vgl. [2], S.34)

2.3.2.2 Wichtige Tabellen in der TYPO3-Datenbank

Folgende Tabelle gibt einen Überblick über die wichtigsten Tabellen der TYPO3-Datenbank, betreffend der zu programmierenden Sucherweiterung:

Tabelle	Beschreibung
pages	In dieser Tabelle befinden sich alle Seiten der verwalteten Webpräsenz.
tt_content	Die Tabelle beinhaltet die Inhaltselemente der einzelnen Seiten.

Tabelle 1: Wichtige Tabellen in der TYPO3-Datenbank (vgl. [2], S.35)

2.3.3 Verzeichnisstruktur

Eine TYPO3-Standardinstallation liefert folgende Verzeichnisstruktur (gekürzt auf die wesentlichen Verzeichnisse betreffend der zu programmierenden Sucherweiterung):

Verzeichnis	Beschreibung
t3lib/	Klassenbibliotheken, die von Modulen, Extensions und Funktionen verwendet werden können
typo3/	Beinhaltet das TYPO3-Backend
ext/	Globale Extensions
sysext/	System-Extensions
cms/	Extension „CMS“
tslib/	Klassenbibliothek zur Erzeugung der Frontend-Ausgabe
typo3conf/	Lokale Extensions
uploads/	Von Benutzerinnen/Benutzern hochgeladene Dateien

Tabelle 2: Auszug aus der TYPO3-Verzeichnisstruktur (vgl. [2], S. 37)

2.4 Das Backend

Das TYPO3-Backend ist die zentrale Verwaltungsstelle des Systems.

Das TYPO3-Backend gliedert sich standardmäßig in 2 Bereiche:

- **Moduleliste** auf der linken Seite
- **Arbeitsbereich** auf der rechten Seite

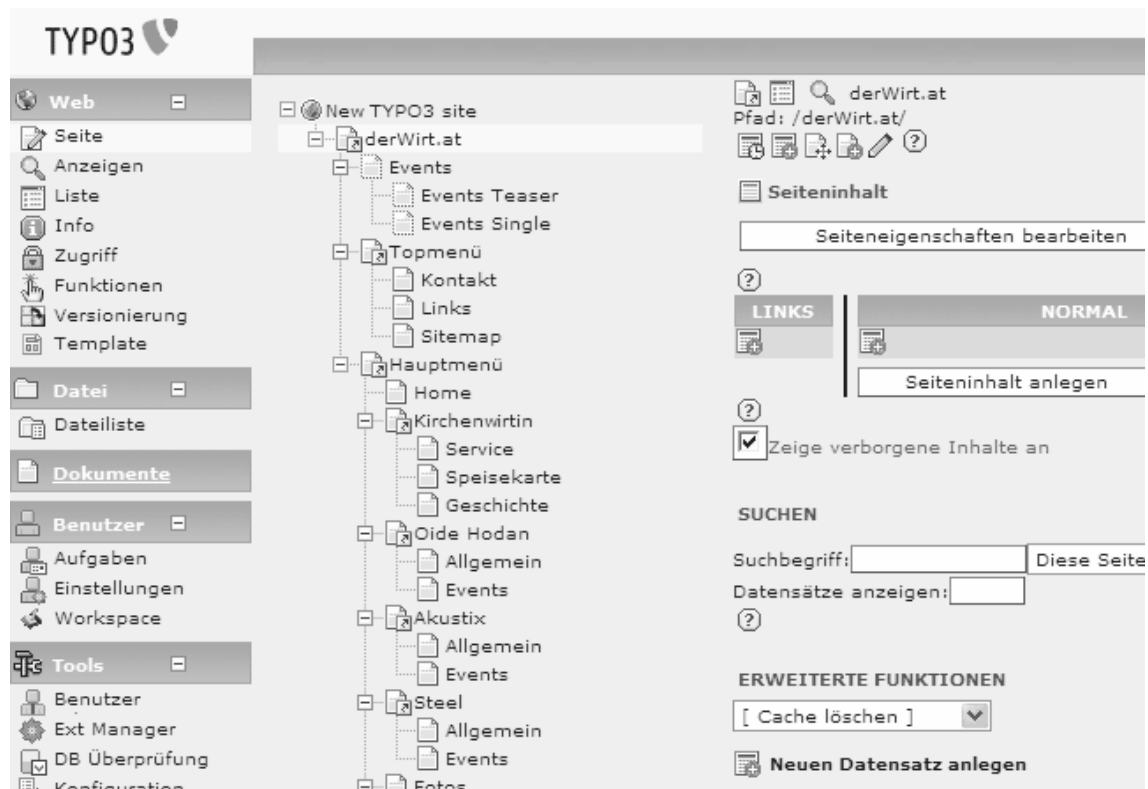


Abbildung 3: Das TYPO3-Backend

TYPO3 ist modular aufgebaut. Die Module im Backend sind alle über Einzelbefehle zugänglich. Die Befehle sind in zusammengehörige Gruppen zusammengefasst, die alle über die Modulleiste erreichbar sind. (vgl. [1], S.457)

Die Modulleiste ist standardmäßig in folgende Modulgruppen gegliedert:

- **Web**
Das Modul „Web“ beinhaltet die wichtigsten Funktionen für das Erstellen, Betrachten und Bearbeiten von Seiten.
- **Datei**
Das Modul „Datei“ dient zur Dateiverwaltung innerhalb des CMS.
- **Dokumente**
Ein Modul, das das aktuell bearbeitete Dokument im Arbeitsbereich öffnet.
- **Benutzer**
Das Modul „Benutzer“ stellt Funktionen betreffend Workflow, Benutzerprofil-Einstellungen und Workspace-Einstellungen bereit.

- **Tools**
Eine Modulgruppe, die Funktionen zur Systemkonfiguration, -überprüfung und -verwaltung anbietet.
- **Hilfe**
Dieser Bereich liefert Infos zu den einzelnen Modulen sowie einen Link zum TYPO3-Onlinehandbuch.
- **Admin Funktionen**
Die „Admin Funktionen“ ermöglichen das Löschen des Datencaches.
Diese Funktionen sind standardmäßig nur dem Administrator zugänglich.

(vgl. [1], S. 457ff)

2.5 TypeScript

TypeScript ist keine Programmier- oder Scriptsprache, es ist eine Konfigurationssprache, die in TYPO3 Anwendung findet.

TypeScript wird für folgende Aufgabenbereiche verwendet:

- **TypeScript Templates**
TypeScript wird zur Definition der Templates, die als Vorlage für die einzelnen Seiten der Webseite dienen, verwendet.
- **Page TSconfig**
Mittels TSconfig ist es möglich, das TYPO3-Backend und dessen Funktionen über TypeScript-Anweisungen zu gestalten bzw. zu steuern.
- **User TSconfig**
TypeScript dient auch zur individuellen Konfiguration von bestimmten nutzerspezifischen Einstellungen. Es können dazu pro Nutzerin/Nutzer sowie pro Nutzerinnengruppe/Nutzergruppe spezielle Anweisungen festgelegt werden.

(vgl. [1], S. 365f)

2.6 Designvorlagen

Eine Alternative zu den reinen TypoScript-Templates stellen HTML-Designvorlagen dar.

Das endgültige Ausgabeergebnis von TYPO3 ist eine HTML-Seite. Dieser HTML-Code wird auf Grund von TypoScript-Konfigurationen generiert. Neben der Möglichkeit, das gesamte Template per TypoScript zu erstellen, bietet sich auch die Möglichkeit eine extern erstellte HTML-Vorlage in die TypoScript-Konfiguration einzubinden. Diese Option ist der reinen Template-Erstellung per TypoScript vorzuziehen, da sie einem ermöglicht, die Vorlagen komfortabel und unabhängig von TYPO3 in jedem beliebigen Editor zu erstellen. (vgl. [1], S. 187)

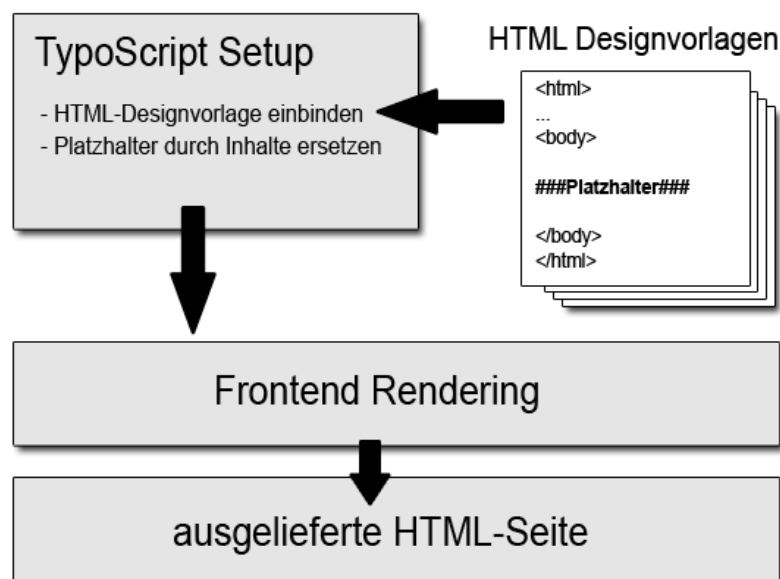


Abbildung 4: Einbindung von HTML-Designvorlagen in TypoScript

Statische Bereiche können fix in die Vorlage integriert werden. Dynamische Bereiche wie z.B. Inhaltsbereich oder auch Menü werden mit Platzhaltern versehen. Die fertigen HTML-Vorlagen werden dann im TypoScript-Setup geladen. Anschließend müssen noch die Platzhalter per TypoScript durch die tatsächlichen Inhalte ersetzt werden. Die fertige Konfiguration wird an die entsprechenden

PHP-Skripte übergeben. Diese Skripte stoßen den Frontend-Rendering-Prozess an. Ergebnis dieses Prozesses ist die fertige HTML-Seite (siehe auch Kapitel 3.3.1).

Die meisten Erweiterungen, die für TYPO3 zur Verfügung stehen, basieren auf dem Designvorlagen-Konzept, da es die Flexibilität des Layouts erhöht und die Vorlagen unabhängig von TYPO3 verwendet werden können.

2.7 Caching

TYP03 erzeugt die einzelnen Seiten, die im Browser dargestellt werden, dynamisch. Bei dieser dynamischen Erzeugung handelt es sich um einen rechenintensiven Prozess. Neben der Eintragung von TypoScript-Befehlen in ein PHP-Array wird ein Baum aus Objekten aufgebaut. Mit Hilfe dieser Objekte wird dann schließlich die auszugebende HTML-Seite gerendert. Um zu verhindern, dass dieser aufwendige Prozess bei jedem Aufruf einer Seite erneut durchgeführt wird, und somit viel Arbeitsspeicher und Prozessorkapazität verbraucht wird, hat man ein ausgereiftes Caching-Konzept entwickelt.

Bei aktiviertem Caching werden statische Seiten nach ihrer Erzeugung als HTML-Code in der Datenbank gespeichert. Werden diese Seiten erneut aufgerufen, wird der bereits fertige HTML-Code, der sich in der Datenbank befindet, ausgeliefert, ohne den ganzen Rendering-Prozess erneut anzustoßen. Gecacht werden neben den bereits genannten statischen HTML-Seiten auch verschiedene Konfigurationseinstellungen, Bilder-Abmessungen, Objektbäume etc. – teils in Tabellen der Datenbank aber auch als Dateien im Dateisystem. Caching bringt jedoch nicht nur Vorteile mit sich. So kann es vorkommen, dass veraltete Inhalte angezeigt werden, da der Cache nicht automatisch aktualisiert wurde. In solchen Fällen muss der Cache gezielt manuell geleert werden.

(vgl. [2], S. 707ff)

2.8 Community

„Um TYPO3 hat sich eine große und lebendige Entwicklergemeinde gebildet – ständig werden neue Verbesserungen und Extensions publiziert. In zahlreichen Foren, Newsgroups und Mailinglisten werden Fragen rund um TYPO3 gestellt und (meistens) beantwortet.“ ([2], S. 27)

Ein wichtiger Anlaufpunkt ist die Webseite <http://www.typo3.org>. Hier findet man neben dem Dokumenten-Archiv das unverzichtbare Extension-Repository (siehe Kapitel 3.1.6). Außerdem kann man sich über die Webseite in eine Vielzahl von Mailinglisten eintragen.

Alternativ gibt es einige Community-Foren, die sich ausschließlich mit Fragen zu TYPO3 beschäftigen. Das größte deutschsprachige Forum findet man unter <http://www.typo3.net>. Etwas jünger ist die Plattform <http://www.typo3forum.net>. (vgl. [2], S. 27)

Zusätzlich zu den Online-Quellen gibt es auch ein eigenes deutschsprachiges TYPO3 Magazin – T3N. T3N steht für TYPO3-News und ist eine monatlich erscheinende Fachzeitschrift. Nähere Informationen findet man unter <http://t3n.yeebase.com>.

Im November 2004 wurde die so genannte TYPO3-Association gegründet. Ziel dieser Organisation ist die technische und organisatorische Weiterentwicklung von TYPO3. Es besteht die Möglichkeit, durch Zahlung eines Mitgliedsbeitrages die Weiterentwicklung des Projektes zu unterstützen. Weitere Informationen zur Organisation gibt es auf der offiziellen Webseite <http://association.typo3.org>. (vgl. [2], S. 27)

2.9 Weitere Features

TYPO3 bringt eine Reihe weiterer Funktionen mit sich. Nachfolgend eine Auflistung der wichtigsten Features:

- **Mehrsprachigkeit**

TYPO3 unterstützt alternative Sprachen zur Standardsprache Englisch sowohl im Backend als auch im Frontend.

- **WYSIWYG-Editoren**

WYSIWYG Editoren dienen zum Editieren der Webseiten-Inhalte. Als Standard verwendet TYPO3 den Editor „HTMLarea“. Es können aber auch alternative Editoren verwendet werden.

- **Grafikmanipulation**

TYPO3 bringt die Fähigkeit mit sich, Grafiken zu erzeugen und zu manipulieren. Dieses Feature wird ua. zur Erzeugung von grafischen Menüs oder zur Erzeugung von Thumbnails verwendet.

- **Seitenbaum**

In TYPO3 werden die einzelnen Seiten einer Webseite in einer übersichtlichen hierarchischen Baumstruktur angezeigt. Diese Darstellung ermöglicht auch bei umfangreicheren Projekten ein rasches Zurechtfinden.

- **Frontend-Editing**

Dieses Feature ermöglicht es der Benutzerin/dem Benutzer direkt im Frontend der Webseite Änderungen durchzuführen. Es ist kein Zugriff auf das Backend nötig.

- **Benutzerverwaltung und Rechtevergabe**

Es besteht die Möglichkeit, die Zugriffsrechte der einzelnen Benutzerinnen/Benutzer detailliert zu steuern.

- **Geschützte Bereiche**

Über die Benutzerverwaltung besteht außerdem die Möglichkeit, geschützte Bereiche im Frontend der Webseite festzulegen, auf die nur berechnigte Benutzerinnen/Benutzer zugreifen können.

- **Workflow-Management**

Durch ein integriertes Workflow-Management ist es möglich, Vorgehens-

weisen für bestimmte Aufgaben festzulegen. Jedoch steckt das Workflow-Management von TYPO3 noch in Kinderschuhen und wird erst in den kommenden Versionen kontinuierlich erweitert werden.

- **Persönliche Aufgaben**

Über das Aufgaben-Modul können die Benutzerinnen/Benutzer untereinander kommunizieren. Es können Aufgaben zugeordnet sowie persönliche Nachrichten ausgetauscht werden.

- **Dynamische und statische Auslieferung**

TYPO3 liefert standardmäßig alle Seiten dynamisch aus. Man kann den Auslieferungsprozess aber auch so konfigurieren, dass die Seiten statisch ausgeliefert werden.

- **Versionsmanagement**

Das in TYPO3 integrierte Versionsmanagement bietet einem die Möglichkeit, Änderungen jederzeit rückgängig zu machen.

(vgl. [2], S. 24ff)

3 TYPO3 Extension-Programmierung

In diesem Kapitel wird auf die Extension-Programmierung in TYPO3 eingegangen. Es werden Grundlagen erläutert, die für die Erstellung einer Erweiterung wichtig sind. Außerdem werden einige interessante Möglichkeiten, die TYPO3 für die Extension-Programmierung anbietet, angesprochen. Abschließend wird die Erweiterung „Kickstarter“ vorgestellt.

3.1 Grundlagen

„Im Allgemeinen versteht man unter einer Extension eine funktionale Erweiterung von TYPO3.“ ([2], S. 368)

Der größte Teil der TYPO3-Funktionalität ist in Erweiterungen untergebracht. Auch die Standardinstallation von TYPO3 bringt bereits ein umfangreiches Paket an Erweiterungen mit sich. (vgl. [1], S. 303)

3.1.1 Namenskonventionen

Jede Extension, die man veröffentlichen will, muss einen weltweit eindeutigen Namen tragen. Diesen Namen bezeichnet man in Anlehnung an primäre Schlüssel in Datenbanken als „Extension-Key“. (vgl. [2], S.513)

Daher sollte die erste Tätigkeit vor der Programmierung einer eigenen TYPO3-Erweiterung die Registrierung eines Extension-Keys sein. Die Registrierung erfolgt über die offizielle TYPO3-Webseite www.typo3.org.

Konventionen bei der Erstellung von TYPO3-Erweiterungen:

- Erlaubt sind die Zeichen „a-z“ und „0-9“ sowie der Unterstrich. Außerdem ist es untersagt, den Namen mit „tx“ oder „u“ beginnen zu lassen.
- Datenbanktabellen, die für eine Extension angelegt werden, wird ein „tx_“ vorangestellt, danach folgt der Extension Key ohne Unterstriche und danach ein frei wählbarer Name.

(vgl. [2], S.513f)

3.1.2 Entwicklungsstatus einer Erweiterung

Jede entwickelte Erweiterung muss mit einem so genannten Entwicklungsstatus versehen werden. Dieser gibt Aufschluss über den Entwicklungsfortschritt der jeweiligen Erweiterung.

Man unterscheidet folgende Zustände:

- **alpha**
Der Status „alpha“ bedeutet, dass sich die Entwicklung noch im Anfangsstadium befindet.
- **beta**
Der Status „beta“ gibt an, dass die Erweiterung noch in Entwicklung ist, jedoch bereits ein Großteil der Funktionalität vorhanden ist.
- **stable**
„stable“ bedeutet, dass die Erweiterung stabil läuft und sich auch bereits in der Praxis bewährt hat.
- **experimental**
Bei einer Erweiterung mit dem Status „experimental“ handelt es sich um eine experimentelle Entwicklung. Es ist offen, ob die Erweiterung fertig gestellt wird.
- **test**
Dient eine Erweiterung zur Demonstration gewisser Konzepte oder rein zu Testzwecken, wird sie mit dem Status „test“ veröffentlicht.

3.1.3 Einteilung von Erweiterungen

Erweiterungen können in 3 Hauptgruppen gegliedert werden:

- **Systemextensions**

Hierbei handelt es sich um Erweiterungen für den Systemkern. Diese befinden sich im Verzeichnis „typo3/sysexst/“.

- **Module**

Bei Modulen handelt es sich um Erweiterungen für das Backend von TYPO3.

- **Plugins**

Von Plugins spricht man im Zusammenhang mit Erweiterungen für das Frontend der Webseite.

Module und Plugins lassen sich weiters unterscheiden, ob sie global oder lokal installiert werden. Werden Module bzw. Plugins global installiert, befinden sie sich im Verzeichnis „typo3/ext/“ und stehen global allen TYPO3-Installationen zur Verfügung. Werden sie hingegen lokal installiert, befinden sich die Erweiterungen im Verzeichnis „typo3conf/ext“ und stehen individuell nur der jeweiligen Installation zur Verfügung.

(vgl. [2], S. 33)

3.1.4 Dateistruktur einer Erweiterung

Meistens beinhaltet eine TYPO3-Erweiterung folgende Dateien:

- **ext_emconf.php**

Diese Datei enthält allgemeine Informationen wie Namen, Kategorie, Status der Erweiterung.

- **ext_localconf.php**

In der Datei findet man hauptsächlich Angaben über zu ladende Klassen.

- **ext_tables.php**

In der „ext_tables.php“ findet man Erweiterungen des „\$TCA“ (siehe auch Kapitel 3.2.2).

- **ext_tables.sql**
Diese Datei beinhaltet SQL-Befehle für die Erzeugung und Erweiterung von Datenbanktabellen.
- **tca.php**
In der „tca.php“ findet man Anpassungen und Änderungen des „\$TCA“ (siehe auch Kapitel 3.2.2).
- **locallang*.php/locallang*.xml**
In dieser Datei werden Sprachinformationen für mehrsprachige Ausgaben gespeichert.
- **ext_icon.gif**
Das „ext_icon.gif“ ist ein 16x16 bzw. 16x18 Pixel großes GIF-Bild. Es wird als Erweiterungs-Icon im Backend verwendet.

Typische Unterordner sind:

- **pi1, pi2**
Diese Ordner werden für die Plugin-Programmierung verwendet.
- **mod1, mod2**
In den „mod“-Ordnern befinden sich die Dateien der einzelnen Module, die die jeweilige Erweiterung beinhaltet.
- **res**
Im „res“-Ordner befinden sich Ressourcen der Erweiterung (Bilder, Stylesheets, Templates usw.).
- **doc**
Der „doc“-Ordner beinhaltet - falls vorhanden - das Handbuch zur Erweiterung.

(vgl. [2], S.514f)

3.1.5 Der Extension-Manager

„Der Extension-Manager ist für die Zusammenarbeit von TYPO3 mit Extensions verantwortlich. Er sorgt bei jedem Aufruf von TYPO3 sowohl im Backend als auch im Frontend dafür, dass die installierten Extensions geladen werden und zur Verfügung stehen.“
([2], S. 516)

Gespeichert werden die installierten Erweiterungen in der Datei „typo3conf/localconf.php“ in dem Array „\$TYPO3_CONF_VARS['EXT']['extList']“. Die Reihenfolge der gespeicherten Einträge ist auch ausschlaggebend für die Ladereihenfolge der Erweiterungen. Ist eine Erweiterung zwar im jeweiligen Ordner abgelegt, jedoch nicht in diesem Array eingetragen, wird sie nicht geladen. (vgl. [2], S. 516).

Die erste Datei, die der Extension-Manager aufruft ist die Datei „ext_emconf.php“ im Ordner der jeweiligen Erweiterung. Danach werden die aufgelisteten Dateien in Kapitel 3.1.4 - falls vorhanden - der Reihe nach geladen.

3.1.6 Online-Extension-Repository

Es gibt weltweit zahlreiche aktive TYPO3-Entwicklerinnen/Entwickler, die ständig neue Erweiterungen implementieren. Diese Erweiterungen sind meist für jederfrau/jedermann über das Online-Extension-Repository zugänglich. Die Online-Version des Extension-Repositorys ist unter <http://typo3.org/extensions/> zu finden.

„Mit Hilfe des Online Extension Repository bietet TYPO3 eine Möglichkeit, die dort zur Verfügung gestellten Extensions auf den eigenen Server herunterzuladen bzw. in das eigene System zu importieren.“ ([2], S. 382)

Für die Zukunft ist geplant, eine Sicherheitskontrolle bei veröffentlichten Erweiterungen durchzuführen. Dazu wurde ein so genanntes „Security-Team“ gebildet. Jede mit dem Status „stable“ bzw. „beta“ veröffentlichte Erweiterung wird

von mindestens 2 Personen des Security Teams auf Sicherheitsmängel überprüft. Befinden diese 2 Personen die Extension für unsicher, wird sie zwar in das Repository aufgenommen, jedoch als „unsupported“ gekennzeichnet. Alle Erweiterungen, die die Sicherheitskontrolle überstehen, werden als „supported Extension“ in die offizielle TYPO3-Extension-Kollektion aufgenommen. [8]

3.1.7 Das Modul „Extension Manager“

Im TYPO3-Backend gibt es eine zentrale Stelle, über die alle Erweiterungen verwaltet werden. Dieses Modul trägt die Bezeichnung „Extension Manager“.

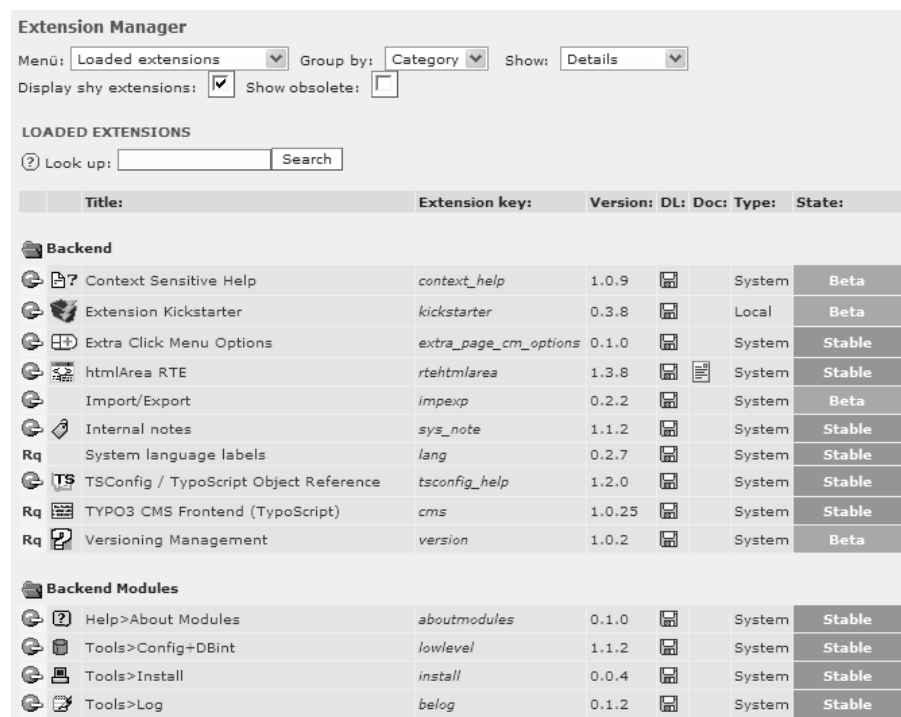


Abbildung 5: TYPO3-Modul „Extension Manager“

Über diesen Manager können Erweiterungen importiert, installiert oder auch wieder entfernt werden. Außerdem kann eine Verbindung zum Online-Extension-Repository hergestellt werden. Steht die Verbindung, kann man nach verfügbaren Erweiterungen suchen und diese direkt installieren. Auch der Veröffentlichungsvorgang einer Erweiterung wird über das Modul „Extension Manager“ abgewickelt. (vgl. [1], S. 303ff)

3.2 Backend-Programmierung

„Das Backend von TYPO3 dient zur Dateneingabe und -pflege. Für den Entwickler von Extensions stellt es umfangreiche Schnittstellen zur Verfügung, die es ermöglichen, Konfigurationen zu ändern und Eingabemasken hinzuzufügen.“ ([2], S.533)

3.2.1 Eingabemasken

Die Eingabe von Daten erfolgt im Backend über Eingabemasken in Form von Formularen. TYPO3 bietet für die Generierung solcher Eingabemasken eigene Funktionen an, auf die auch Erweiterungen zurückgreifen können und sollten. Bei den Eingabemasken handelt es sich um keine vorgefertigten starren Formulare, die auf einzelnen Tabellen basieren. Die Erzeugung erfolgt dynamisch auf Grund eines speziellen Tabellenkonfigurationsarrays – dem „\$TCA“ (siehe auch Kapitel 3.2.2).

3.2.2 Das Table Configuration Array „\$TCA“

„Das \$TCA ist ein globales Array, das alle Informationen zur Erstellung eines Eingabeformulars und zur Bearbeitung von Daten beinhaltet.“ ([2], S. 534)

Die Generierung von Formularen aus dem „\$TCA“ wird von der TYPO3-Core-Engine (TCE), die zum Verarbeiten und Speichern von Daten dient, übernommen. Jeder Eintrag im „\$TCA“ ist an eine Datenbanktabelle geknüpft. Der erste Index des „\$TCA“ besteht immer aus dem Tabellennamen der jeweiligen Tabelle. Die Konfiguration des „\$TCA“ bestimmt die Ausgabe der Eingabemaske im Backend von TYPO3. (vgl. [2], S. 534ff)

3.2.3 Module

„Unter einem Modul wird in TYPO3 eine Backend-Erweiterung verstanden, also eine Erweiterung, die nicht auf der Webseite direkt eingesetzt wird.“ ([3], S.252)

Module sind eigenständig lauffähige PHP-Dateien. Sie befinden sich in Unterordnern der jeweiligen Erweiterung.

Standarddateien im Modulordner:

- **conf.php**
„conf.php“ stellt die Konfigurationsdatei des Moduls dar.
- **index.php**
Die „index.php“ ist Hauptdatei des Backend-Moduls. Sie enthält das lauffähige Modul.
- **locallang.php**
Die Datei „locallang.php“ enthält die verschiedenen Sprachversionen für das Modul.
- **locallang_mod.php**
In der „locallang_mod.php“ findet man die verschiedenen Sprachversionen für die Modulleiste.
- **moduleicon.gif**
„moduleicon.gif“ stellt das Icon dar, das in der Modulleiste angezeigt wird.

(vgl. [2], S. 560ff)

3.3 Frontend-Programmierung

3.3.1 Das Frontend-Rendering

Unter Frontend-Rendering versteht man die Erzeugung des Frontend-Quellcodes. Der TYPO3-Renderingprozess findet in der Datei „tslib/index_ts.php“ statt. Der Ordner „tslib“ gehört zur Systemextension „CMS“, sprich das Frontend-Rendering wird von dieser Erweiterung ausgeführt. (vgl. [2], S. 577)

3.3.1.1 „\$TSFE“ (TypoScript-Frontend)

Die Hauptarbeit im Renderingprozess übernimmt die Klasse „tslib_fe“, die global als Objekt in „\$TSFE“ (TypoScript-Frontend) zur Verfügung steht. „\$TSFE“ stellt ein globales Objekt dar, dass wichtige Informationen für das TYPO3-Frontend enthält. Im „\$TSFE“ sind einige Klasseninstanzen als Objekte vorhanden. Auf diese kann in Plugins zurückgegriffen werden. (vgl. [2], S. 577)

Objekt	Beschreibung
\$TSFE- > sys_page	Zuständig für den Aufruf der Seite
\$TSFE- > fe_user	Login- und Sessioninformationen für Frontend-Benutzerinnen/Benutzer
\$TSFE- > ttmpl	Methoden zur Bearbeitung von TypoScript-Templates
\$TSFE- > cObj	Verantwortlich für das Frontend-Rendering der Inhaltselemente
\$TSFE- > csConvObj	Zuständig für die Umwandlung von Charakter-Sets

Tabelle 3: Objekte im „\$TSFE“ (vgl. [2], S. 579)

3.3.1.2 Der Rendering-Prozess

„\$TSFE“ werden als erstes die grundlegenden Übergabeparameter wie z.B. die „id“ der aufzurufenden Seite übergeben. Als nächstes werden Backend- und Frontend-Benutzerin/Benutzer auf Berechtigung geprüft. Auf Grund der übergebenen Parameter werden nun die Inhalte des entsprechenden Eintrags der Tabelle „pages“ als Array abgelegt. Im nächsten Schritt wird das „\$TCA“ geladen. Es beinhaltet alle Tabellen der Datenbank und steht global für das Frontend-Editing zur Verfügung. Nächste Aufgabe im Rendering-Prozess ist die Verarbeitung des TypoScript-Codes.

Ist das Caching aktiviert, wird nun in der entsprechenden Datenbanktabelle überprüft, ob die aktuelle Seite bereits vorhanden ist. Ist dies der Fall, wird das fertige HTML Konstrukt der Seite in das Objekt „\$TSFE->content“ geladen.

Wurde die Seite noch nicht gecached bzw. ist das Caching deaktiviert, wird im nächsten Schritt die Seitenerzeugungsdatei „tslib/pagegen.php“ inkludiert. Diese Datei ist zuständig für die Generierung der Seite aus ihren Content-Objekten entsprechend der Vorgaben im TypoScript-Code. Es wird zuerst ein Objekt der Klasse „tslib_content“ in „\$TSFE->cObj“ erzeugt. Dieses Objekt ist für die Frontend-Programmierung sehr wichtig, da es die Inhalte der Content-Objekte umfasst und diese rendert. Ist die Ausführung der Seitenerzeugungsdatei abgeschlossen, steht der Seiteninhalt in der Variablen „\$TSFE->content“ zur Verfügung. Abschließend erfolgt die Headerausgabe und danach die Ausgabe von „\$TSFE->content“ – die Frontend-Seite ist erzeugt.

(vgl. [2], S. 577ff)

3.3.2 Plugins

„Unter einem Plugin wird in TYPO3 eine Frontend-Erweiterung verstanden. Frontend-Plugins können direkt auf der Webseite eingesetzt werden und stehen im Regelfall im Seiteninhaltstyp Plugin einfügen in einer Auswahl zur Verfügung.“ ([3], 2006, S. 252)

Die Standardklasse für Plugins ist die Klasse „tslib_pibase“. Die meisten Plugins stellen eine Erweiterung dieser Klasse dar. Die Klasse enthält bereits eine Vielzahl praktischer Funktionen, die einem jeden Plugin zur Verfügung stehen. Zu finden ist die Klasse in der Datei „tslib/class.tslib_pibase.php“.

Die Klasse „tslib_pibase“ stellt Methoden zur Verfügung, die wie folgt eingeteilt werden können:

- **Link-Methoden**

Bei Hyperlinks im Frontend ist zu beachten, dass diese den unterschiedlichen Einsatzmöglichkeiten von TYPO3 (z.B. Mehrsprachigkeit) entsprechen. Diese Funktionalität bringen die Link-Methoden der Klasse „tslib_pibase“ mit sich. Es gibt eine Vielzahl vordefinierter Funktionen, die die Besonderheiten von TYPO3 berücksichtigen und somit für die Plugin-Programmierung unumgänglich sind.

- **Auflistung von Datensätzen**

Für die Auflistung von Datensätzen stehen wiederum eine Reihe vorhandener Funktionen zur Verfügung, die der Entwicklerin/dem Entwickler einigen Arbeitsaufwand ersparen.

- **Stylesheets und CSS**

Zur Erleichterung des Umgangs mit Stylesheets und CSS werden auch für diesen Bereich diverse Funktionen angeboten.

- **Frontend-Editing**

Es werden Methoden zur Verfügung gestellt, die für die Ausgabe der Konfigurationsleisten für das Frontend-Editing zuständig sind. Diese Konfigurationsleisten können somit auch problemlos in Plugins eingebunden werden.

- **Mehrsprachigkeit und Lokalisation**

Für die Erzeugung mehrsprachiger Ausgaben werden „Locallang-Dateien“ verwendet. Es werden Methoden benötigt, die zum Aufrufen der Texte aus diesen Sprachdateien dienen. Diese Methoden werden ebenfalls von der Klasse „tslib_pibase“ zur Verfügung gestellt.

- **Methoden für Datenbankabfragen**

Um eine TYPO3-konforme Datenbehandlung zu erreichen, werden Methoden für Datenbankabfragen angeboten.

- **Methoden für Flexforms**

Es werden Methoden zur Verfügung gestellt, die speziell auf die Behandlung von im „Flexform-Prinzip“ gespeicherter Daten ausgerichtet sind.

(vgl. [2], S. 597ff)

3.4 Flexforms

Im Unterschied zur traditionellen Dateneingabe im Backend, bei der jedes Formularfeld auch einem Feld in der Datenbank entspricht, werden bei Flexforms die Eingaben verschiedener Eingabefelder gemeinsam in einem einzigen Datenbankfeld gespeichert. Dieses Datenbankfeld befindet sich in der Tabelle „tt_content“ und trägt die Bezeichnung „pi_flexform“. Strukturiert werden die einzelnen Eingaben mittels XML, wobei unterschiedliche Plugins auch unterschiedliche Konfigurationseingaben benötigen. Dies spiegelt sich in einer eigenen XML-Struktur wider. In TYPO3 bezeichnet man diese Strukturierung „data structure“. (vgl. [2], S. 608)

„Flexforms werden also u.a. deshalb als flexibel betrachtet, weil für unterschiedliche Plugins unterschiedlichste Feldkombinationen (data structures) abgespeichert werden können, ohne dass deshalb die Datenbanktabelle geändert werden müsste.“ ([2], S. 608)

Die „data structures“ werden mittels einer eigenständigen XML-Datei festgelegt. Diese Datei wird dann in das „\$TCA“ mittels Referenzierung eingebunden. (vgl. [2], S. 614)

3.5 Mehrsprachigkeit

Jede Benutzerin/Jeder Benutzer hat die Möglichkeit, die Sprache des Backends frei zu wählen. Natürlich gibt es im Frontend ebenfalls die Option, mehrsprachig zu arbeiten. (vgl. [2], S. 633)

„Auch Extensions können und sollen für Mehrsprachigkeit kompatibel programmiert werden. Das ist insbesondere dann wichtig, wenn Sie sie veröffentlichen möchten.“ ([2], S. 633)

Die unterschiedlichen Sprachversionen müssen dazu in so genannten „Local-lang-Dateien“ gespeichert werden. Der Dateiname einer solchen Datei beginnt immer mit „locallang“. Das Dateiformat kann zwischen PHP und XML gewählt werden. Die Sprachdatei enthält in Normalfall alle Sprachen, wobei die „Standard-Sprache“ immer Englisch ist. Dies ist vor allem wichtig, wenn keine Übersetzung für die aktuell eingestellte Sprache gefunden wird, da in diesem Fall immer auf die „Standard-Sprache“ zurückgegriffen wird. (vgl. [2], S. 633ff)

3.6 Extension „Kickstarter“

Mit der Extension „Kickstarter“ (Extension-Key „kickstarter“) können schnell und komfortabel die benötigten Dateien und Schnittstellen für eine eigene TYPO3-Erweiterung erstellt werden. Es werden Dateien erzeugt, die die Grundkonfiguration beinhalten, die man über den „Kickstarter“ angegeben hat. (vgl. [3], S. 302)

Man kann den „Kickstarter“ als Standardwerkzeug für die Entwicklung einer Erweiterung unter TYPO3 bezeichnen, da er die Arbeit der Entwicklerin/des Entwicklers erheblich erleichtert. Neben einer Oberfläche zur „\$TCA“-Gestaltung verfügt er über die Möglichkeit, bereits verwendbare Plugins und Module bereitzustellen. Diese müssen dann „nur noch“ mit den eigentlichen Funktionalitäten gefüllt werden. (vgl. [2], S. 658)

4 Erweiterung planen

Um mit der Entwicklung der Erweiterung zu beginnen, muss zuerst definiert werden, welche Anforderungen das fertige Softwareprodukt erfüllen soll. Zusätzlich müssen bestehende Sucherweiterungen auf Synergien hin überprüft werden. Schlussendlich muss dann entschieden werden, ob Synergien bzw. Vorgehensweisen einer bestehenden Erweiterung genutzt werden können.

4.1 Anforderungsdefinition

Folgende Anforderungen soll die zu programmierende Sucherweiterung erfüllen:

4.1.1 Was soll durchsucht werden?

Die Sucherweiterung dient zum Durchsuchen des Inhalts einer jeden TYPO3-Webseite. Die Erweiterung soll in den Standard-Inhaltselementen von TYPO3 sowie in den Inhalten jeder beliebigen Extension suchen.

4.1.2 Reihung der Ergebnisse

Die Reihung der Suchtreffer soll auf Grund folgender Kriterien erfolgen:

- **Bewertungen**

Die Benutzerinnen/Benutzer sollen die Möglichkeit haben, die Suchtreffer nach deren Relevanz zu bewerten. Positive Bewertungen heben den Rang des Treffers, negative Bewertungen senken den Rang des Treffers.

- **Anzahl der Klicks**

Ein Klick auf ein Suchergebnis soll sich positiv auf dessen Position in der Reihung der Ergebnisse auswirken.

- **Anzahl der Treffer in einem Ergebnis**

Es soll außerdem berücksichtigt werden, wie oft ein gesuchter Begriff im jeweiligen Ergebnis vorkommt. Mehrere Übereinstimmungen sollen sich positiv auf die Position des Suchtreffers auswirken.

Diese Kriterien sollen mit einer Gewichtung versehen werden. Diese Gewichtung soll von der Benutzerin/vom Benutzer manuell konfigurierbar sein.

4.1.3 Spracheinstellungen

Die Erweiterung soll auf Mehrsprachigkeit ausgerichtet sein (siehe auch Kapitel 3.5).

4.1.4 Templates

Die Erweiterung soll auf Grund höherer Flexibilität basierend auf externen Designvorlagen erstellt werden.

4.1.5 Statistik

Im Backend soll eine Statistik über die getätigten Suchanfragen und deren Ergebnisse zur Verfügung stehen.

4.1.6 Usability

Die Sucherweiterung soll leicht zu installieren sein. Zusätzlich soll die Erweiterung sowohl im Frontend als auch im Backend intuitiv sein, um die Bedienung so einfach als möglich zu gestalten.

4.1.7 Dokumentation

Um der Benutzerin/dem Benutzer bzw. der Programmiererin/dem Programmierer eine Hilfestellung zur Installation bzw. Adaption der Erweiterung zu liefern, soll ein Handbuch bereitgestellt werden.

4.2 Analyse bestehender Sucherweiterungen

Um eventuelle Synergien bei bereits bestehenden Sucherweiterungen nutzen zu können, müssen die vorhandenen Erweiterungen sondiert und analysiert werden. Zentrale Stelle dieser Sondierung ist das Online-Extension-Repository (siehe auch Kapitel 3.1.6).

Es gibt eine Sucherweiterung, die standardmäßig in der TYPO3-Installation enthalten ist – die „Indexed Search Engine“ (Key: „indexed_search“). Sie ist die am meisten verwendete Sucherweiterung für TYPO3.

Zusätzlich gibt es im Extension-Repository eine Vielzahl von Erweiterungen, die sich mit dem Thema Suche beschäftigen. Daher muss ein Kriterium gefunden werden, welches eine Einschränkung auf 3 zusätzliche Sucherweiterungen, die untersucht werden, ermöglicht. Ein wichtiges Kriterium für eine Erweiterung ist die Verbreitung. Da die Anzahl der Installationen nicht bekannt ist, muss man eine Kennzahl finden, um sich der tatsächlichen Verbreitung anzunähern. Eine Kennzahl, die im Extension-Repository zu jeder Erweiterung angeboten wird, ist die Anzahl der Downloads. Daher liegt es nahe, diesen Wert als Kriterium festzulegen.

Zusätzlich gehören „spezialisierte“ Sucherweiterungen ausgegrenzt. Ziel dieser Diplomarbeit ist es, eine Suche zu implementieren, die den Inhalt einer TYPO3-Webseite durchsucht. Es gibt jedoch auch eine Reihe von Sucherweiterungen, die sich auf das Durchsuchen bestimmter Erweiterungen spezialisiert haben. Diese werden bei der Sondierung nicht berücksichtigt.

Mit dem Kriterium Downloadhäufigkeit ist es nun möglich, die 3 meist heruntergeladenen Sucherweiterungen zu finden. Das Sondieren der Sucherweiterungen im Online-Extension-Repository ergab folgendes Ergebnis (Stand: 16. Juni 2007):

- Extension „**Content Search**“ (Key: „gst_contentsearch”)
1832 Downloads
- Extension „**[AN] Search It**“ (Key: „an_searchit”)
1729 Downloads
- Extension „**Ajax Searching Engine**“ (Key: „mh_ajaxsearch”)
1458 Downloads

4.2.1 Erweiterung „Indexed Search Engine“

Die Extension „Indexed Search Engine“ ist Standardbestandteil der TYPO3-Installation und dadurch die am weitest verbreitete Sucherweiterung. Geschrieben wurde die Erweiterung von „TYPO3-Erfinder“ Kasper Skårhøj. (vgl. [3], S. 288)

Die Erweiterung liefert einerseits eine spezielle Indizierungs-Engine, die die einzelnen Seiten einer Webseite durch einen Indizierungsprozess auflistet. Andererseits liefert die Erweiterung ein Frontend-Plugin, welches ein Suchformular zur Verfügung stellt, mit dem die Webseite nach bestimmten Wörtern durchsucht werden kann. (vgl. [4], S.1)

4.2.1.1 Architektur

Die „Indexed Search Engine“ befindet sich bei einer TYPO3-Installation bereits standardmäßig im Ordner „typo3/sysexst/“ und stellt demnach eine System-Extension dar. (siehe auch Kapitel 3.1.3)

Sie setzt sich wie folgt zusammen:

- **Frontend-Plugin**
Das Frontend-Plugin liefert ein Suchformular, mit dem die Webseite durchsucht werden kann. Es wird eine erweiterte Suche angeboten. Man

kann nach Wörtern oder ganzen Sätzen suchen, Kombinationen von Suchbegriffen durch die Logischen Schlüsselwörter „AND“, „OR“ und „NOT“ bilden. Zusätzlich ist es möglich, die Suche auf bestimmte Bereiche der Webseite einzuschränken. Es können neben normalen HTML-Seiten auch Dokumente (PDF, Word, TXT) oder auch externe URLs durchsucht werden. Das Frontend-Plugin beinhaltet neben dem Suchformular auch gleichzeitig die Ergebnisausgabe. (vgl. [4], S.2f)

- **Modul „Tools > Indexing“**

In diesem Modul findet man Statistiken zur Indizierungs-Engine. Es gliedert sich in 3 Bereiche: „General statistics“, „List: TYPO3 Pages“ und „List: External Documents“. Unter „General statistics“ findet man Informationen über die Anzahl der indizierten Seiten, Wörter, Relationen usw. Die Bereiche „List TYPO3 Pages“ bzw. „List: External Documents“ beinhalten eine Auflistung aller indizierten Seiten bzw. aller externen Dokumente. (vgl. [4], S.7)

- **Modulerweiterung „Web > Info“**

Durch die Modulerweiterung werden dem bereits vorhandenen Modul „Web > Info“ die 2 Bereiche „Indexed Search“ und „Indexed Search statistics“ hinzugefügt. Im Bereich „Indexed Search“ erhält man einen Überblick über alle Seiten der Webseite, ob bereits indiziert oder noch nicht. Bei bereits indizierten Seiten erhält man detaillierte Informationen wie z.B. das Datum der Indizierung, die einzelnen auf der Seite gefundenen Wörter oder auch die Größe der indizierten Seite. Außerdem hat man die Möglichkeit, bereits indizierte Seiten wieder aus dem Index zu löschen. Im Bereich „Indexed Search statistics“ findet man eine Statistik über die meist gesuchten Begriffe. (vgl. [4], S.5f)

4.2.1.2 Technische Details

Kernstück der Erweiterung „Indexed Search Engine“ ist mit Sicherheit die Indizierungs-Engine. Mit ihr werden im Laufe der Zeit alle Seiten der Webseite indiziert, sprich deren einzelne Wörter in eine Datenbanktabelle geschrieben, Relationen zu Seiten gebildet usw. Entscheidend für die Indizierung einer Seite ist,

dass sie gecacht wird. Es werden nämlich ausschließlich gecachte Seiten indiziert. Ist bei einer beliebigen Erweiterung das Caching aktiviert, werden auch deren Inhalte in den Index aufgenommen. Wird eine Seite im Frontend zum ersten Male aufgerufen, wird sie vollständig gerendert und in die Caching-Tabelle eingetragen. Gleichzeitig wird sie für die Sucherweiterung indiziert. Die Seite wird dabei in ihre Bestandteile (Wörter) zerlegt. Die einzelnen Wörter werden mit Relation zur jeweiligen Seite in eine Datenbanktabelle geschrieben, die dann vom Frontend-Plugin durchsucht wird. (vgl. [4], S.8)

4.2.1.3 Datenbanktabellen

Für die Funktion der Erweiterung werden 6 zusätzliche Datenbanktabellen benötigt. (vgl. [4], S.24ff)

4.2.1.4 Reihung der Suchergebnisse

Die Reihung der gefundenen Seiten erfolgt nach folgenden Kriterien:

- Häufigkeit des Wortes auf der Seite
- Abstand des Wortes zum Seitenanfang
- Position des Wortes: title, meta-keywords, meta-description, body

Die Summe dieser Kriterien ergibt dann eine Prozent-Gewichtung und bestimmt die Reihung des Ergebnisses. Zusätzlich zu dieser automatischen Reihung hat die Benutzerin/der Benutzer auch die Möglichkeit, die Resultate nach bestimmten Kriterien zu sortieren.

(vgl. [4], S.26)

4.2.2 Erweiterung „Content Search“

Die Erweiterung „Content Search“ ist im offiziellen Online-Extension-Repository zu finden. Mit 1832 Downloads ist sie die meist verbreitete Sucherweiterung nach der „Indexed Search Engine“. Geschrieben wurde die Erweiterung von Stefan Hafeneger.

4.2.2.1 Architektur

Die Erweiterung wird als lokale Extension im Ordner „typo3conf/ext/“ installiert. (siehe auch Kapitel 3.1.3)

Sie beinhaltet ein Frontend-Plugin, welches ein Suchformular zur Verfügung stellt. Mit diesem Formular kann der Inhalt der Webseite nach Wörtern oder auch ganzen Sätzen durchsucht werden. Eine erweiterte Suche steht nicht zur Verfügung. Zusätzlich zum Suchformular beinhaltet das Plugin auch die Ergebnisausgabe.

4.2.2.2 Technische Details

Der Suchprozess besteht aus einer Datenbankabfrage. Es wird die Tabelle „tt_content“ auf Übereinstimmungen in den Feldern „header“, „bodytext“ und „imagecaption“ durchsucht. Erweiterungen werden also nicht durchsucht. Zusätzlich gibt es einige TypoScript-Konfigurationsmöglichkeiten. Man kann z.B. gewisse Seiten von der Suche ausschließen oder auch die maximale Anzahl der anzuzeigenden Wörter in der Ergebnisliste festlegen.

4.2.2.3 Datenbanktabellen

Die Erweiterung benötigt für ihre Funktion keine zusätzlichen Datenbanktabellen. Es wird ausschließlich die Tabelle „tt_content“ durchsucht.

4.2.2.4 Reihung der Suchergebnisse

Die Suchergebnisse werden nicht nach bestimmten Kriterien sortiert. Entscheidend für die Reihung ist die „id“ in der Datenbank, sprich der Zeitpunkt, zu dem das Inhalts-Element erstellt wurde.

4.2.3 Erweiterung „[AN] Search It“

Die Erweiterung „[AN] Search It“ befindet sich ebenfalls im Online-Extension-Repository und wurde insgesamt bereits 1729 Mal herunter geladen. Geschrieben wurde die Erweiterung von Markus Schmid.

4.2.3.1 Architektur

Die Erweiterung wird als lokale Extension im Ordner „typo3conf/ext“ installiert. (siehe auch Kapitel 3.1.3)

Sie liefert 3 Frontend-Plugins:

- **Formular**

Das „Formular-Plugin“ stellt das Suchformular zur Verfügung. Eine erweiterte Suche wird nicht angeboten.

- **Resultats-Überblick**

Das Plugin liefert eine Liste der „Sections“ (Kategorien), in denen Suchtreffer gefunden wurden.

- **Ergebnisliste**

Das Plugin „Ergebnisliste“ gibt die Ergebnisliste zu den Suchanfragen aus.

(vgl. [5], S. 2)

4.2.3.2 Technische Details

Die Erweiterung wird komplett per TypoScript konfiguriert. Es können verschiedene Tabellen angegeben werden, die durchsucht werden sollen. Die Kombination von einer Tabelle und den entsprechenden Feldnamen wird als „Section“ bezeichnet. Es können verschiedene „Sections“ festgelegt werden, in denen gesucht wird. Neben den normalen Seiteninhalten (Tabelle „tt_content“) können daher auch zusätzlich beliebige Erweiterungen durchsucht werden. (vgl. [5], S. 2)

4.2.3.3 Datenbanktabellen

Für die Funktionalität der Erweiterung werden keine zusätzlichen Tabellen benötigt. Es wird per TypoScript definiert, welche Tabellen und Felder durchsucht werden.

4.2.3.4 Reihung der Suchergebnisse

Die Suchergebnisse werden nicht nach bestimmten Kriterien sortiert. Entscheidend für die Reihung ist die „id“ in der Datenbank.

4.2.4 Erweiterung „AJAX Searching Engine“

Zu finden ist die Erweiterung „AJAX Searching Engine“ über das Online-Extension-Repository. Die Erweiterung, entwickelt von Mario Heim, wurde bereits 1458 Mal herunter geladen.

Wie der Name der Erweiterung bereits verrät, basiert sie auf AJAX (Asynchronous JavaScript and XML). Mittels AJAX können Teile einer Webseite ausgetauscht werden, ohne die gesamte Seite neu zu laden. So kann der Suchvorgang bereits beim Eintippen des ersten Zeichens in das Suchformular im Hintergrund angestoßen werden und gleichzeitig Ergebnisse liefern. Für die Funktionalität der „AJAX Searching Engine“ ist eine zusätzliche Erweiterung namens „XAJAX“ (Key: „xajax“) erforderlich. (vgl. [6], S. 1)

„XAJAX“ liefert eine PHP-Bibliothek, welche die Erstellung von mächtigen, webbasierten AJAX-Applikationen unter der Verwendung von HTML, CSS, JavaScript und PHP ermöglicht. [9]

4.2.4.1 Architektur

Die Erweiterung wird im Ordner „typo3conf/ext“ als lokale Extension installiert. (siehe auch Kapitel 3.1.3)

Die Erweiterung setzt sich wie folgt zusammen:

- **Frontend-Plugin**

Dieses Plugin beinhaltet das Suchformular und die Ausgabe der Ergebnisliste.

- **Backend-Modul „Web > Suchprotokoll“**

Liefert Statistiken über die letzten 15 Suchanfragen und die 15 meist getätigten Suchanfragen.

4.2.4.2 Technische Details

Wie schon erwähnt, basiert die Erweiterung auf AJAX. Es wird bei Eingabe jedes Zeichens in das Suchformular eine Anfrage an die Datenbank geschickt und die entsprechenden Ergebnisse angezeigt, ohne dass die gesamte Seite neu geladen wird.

Standardmäßig wird ausschließlich die Tabelle „tt_content“ durchsucht. Es gibt aber im Backend die Möglichkeit, die zu durchsuchenden Tabellen manuell zu erweitern. So können auch beliebige Erweiterungen durchsucht werden.

4.2.4.3 Datenbanktabellen

Für das Backend-Modul wird eine Datenbanktabelle benötigt, die zu Statistickzwecken die Suchbegriffe, die Anzahl der Resultate und die „id“ der schlussendlich geklickten Seite beinhaltet.

4.2.4.4 Reihung der Suchergebnisse

Die Suchergebnisse werden nicht nach bestimmten Kriterien sortiert. Entscheidend für die Reihung ist die „id“ in der Datenbank.

4.3 Zusammenfassung

Durch die Analyse der verschiedenen Sucherweiterungen sind einige interessante Aspekte in den Vordergrund gerückt, die für die Programmierung der Sucherweiterung wichtig sind.

- **Reihung der Ergebnisse**

Was die Reihung der Suchergebnisse betrifft, ist die „Indexed Search Engine“ die einzige Erweiterung, die durch das Zusammenspiel diverser Kriterien versucht, die Suchergebnisse zu gewichten. Jedoch wird die Meinung der Benutzerin/des Benutzers nicht berücksichtigt. Die Besucherin/der Besucher der Webseite hat keine Möglichkeit den internen Reihungsprozess bewusst zu beeinflussen.

- **Indizierung oder Direktsuche**

Es haben sich 2 verschiedene Suchkonzepte herauskristallisiert. Auf der einen Seite die „Indexed Search Engine“. Diese Erweiterung listet die verschiedenen Inhaltselemente durch ein Indizierungssystem in separaten Datenbanktabellen. Diese Tabellen werden dann vom Frontend-Plugin durchsucht. Die automatische Indizierung der Seiten ermöglicht praktischerweise auch das Durchsuchen beliebiger Erweiterungen.

Auf der anderen Seite stehen die restlichen analysierten Erweiterungen. Sie verzichten auf ein Indizierungssystem. Es werden die jeweiligen Datenbanktabellen – hauptsächlich die Tabelle „tt_content“ - direkt durchsucht. Will man zusätzlich auch diverse Erweiterungen durchsuchen, muss dies manuell nachkonfiguriert werden.

Das Indizierungssystem bietet den Vorteil, dass auch Inhalte von Erweiterungen mitindiziert werden und dafür keine manuelle Konfiguration notwendig ist. Voraussetzung dafür ist jedoch, dass das Caching für alle Seiten und Erweiterungen aktiviert sein muss. Einen erheblichen Nachteil des Indizierungssystems stellt jedoch die Performance dar. Bei umfangreichen Webseiten, bei denen viele Seiten indiziert werden, kann es zu

Performance-Problemen auf Grund vieler Datensätze in den Datenbanktabellen kommen. Dies kann auch in diversen Foren beobachtet werden. Dort werden wiederholt Serverüberlastungen durch die Extension „Indexed Search Engine“ gemeldet. Aus diesem Grund steht die Erweiterung auch derzeit unter Beobachtung. (vgl. [5], S.26)

- **AJAX Technologie**

Die Verwendung von AJAX in der Erweiterung „Ajax Searching Engine“ stellt eine interessante Möglichkeit dar. Applikationen, die mit AJAX realisiert wurden, wirken deutlich interaktiver. Es kann dynamisch auf Eingaben reagiert werden, was spontanere Reaktionen ermöglicht. [10]

4.4 Schlussfolgerung

Unter Berücksichtigung der Überlegungen zu den einzelnen Aspekten erscheinen folgende Schlüsse für die geplante Erweiterung sinnvoll:

- **Direktsuche in der Datenbank**

Das Indizierungsprinzip der „Indexed Search Engine“ wird auf Grund der angesprochenen Performanceprobleme nicht verwendet. Außerdem ist die Erweiterung dadurch unabhängig von den Caching-Einstellungen der jeweiligen TYPO3-Installation.

- **Verwendung von AJAX**

Durch die Verwendung von AJAX kann die Erweiterung interaktiver gestaltet werden.

- **Reihung der Suchergebnisse**

Die Benutzerin/Der Benutzer hat in keiner der analysierten Erweiterungen die Möglichkeit, die Reihung der Ergebnisse bewusst zu beeinflussen. Aus diesem Grund stellt die Implementierung einer Bewertungsmöglichkeit eine Innovation unter den TYPO3-Suchextensions dar.

5 „Intelligent Search Engine“

Dieses Kapitel beschäftigt sich mit der entwickelten TYPO3-Sucherweiterung „Intelligent Search Engine“. Es werden zu Beginn grundlegende Entscheidungen angesprochen. Des Weiteren wird auf die Architektur und vor allem auf die Umsetzung der fertigen Erweiterung eingegangen. Implementierungsschritte, die wesentlich für die Funktion der „Intelligent Search Engine“ sind, werden erläutert. Abschließend wird kurz auf die Veröffentlichung eingegangen.

5.1 Grundlegendes

5.1.1 Entwicklungsumgebung

Für die Entwicklung der Sucherweiterung wird eine lokale Entwicklungsumgebung gewählt. Verwendet wird XAMPP (Apache, MySQL, PHP) in der Version 1.6.2. In dieser Umgebung wird TYPO3 in der Version 4.0.4 installiert.

5.1.2 Extension-Key

Wie bereits in Kapitel 3.1.1 erwähnt, sollte der erste Schritt bei der TYPO3-Extension-Entwicklung die Wahl eines entsprechenden Extension-Keys sein. Bei der zu entwickelnden Sucherweiterung handelt es sich um eine Suche, die durch Interaktion mit der Benutzerin/dem Benutzer laufend neue Informationen zu den einzelnen Suchergebnissen sammelt. Aus diesen gesammelten Informationen wird dann die schlussendliche Reihung der Datensätze bestimmt. Diesen Vorgang der Einbindung der Benutzerin/des Benutzers und die daraus resultierende Reihung kann man durchaus als „intelligenten Vorgang“ bezeichnen.

Aus diesem Grund fiel die Extension-Namenswahl auf „Intelligent Search Engine“. Daraus resultierend wurde der Extension-Key „i_search“ („intelligent_search“) gewählt.

5.1.3 Entwicklungsstatus

Die fertige Sucherweiterung wird zu Beginn mit dem Entwicklungsstatus „beta“ starten, da bereits ein Großteil der Funktionalität vorhanden sein soll. Diese Funktionen können dann laufend verbessert und auch erweitert werden. Daran können sich auch andere Entwicklerinnen/Entwickler der TYPO3-Community beteiligen. (siehe auch Kapitel 3.1.2).

5.1.4 Verwendung der Extension „XAJAX“

Um AJAX nutzen zu können, wird die Erweiterung „XAJAX“ verwendet. Diese Erweiterung ist damit Voraussetzung für die Verwendung der „Intelligent Search Engine“.

5.2 Architektur

Die Erweiterung setzt sich zusammen aus:

- **Frontend-Plugin**
- **Backend-Modul**

Zusätzlich sind für die Funktionalität der Erweiterung 2 zusätzliche Datenbanktabellen erforderlich.

5.2.1 Systementwurf

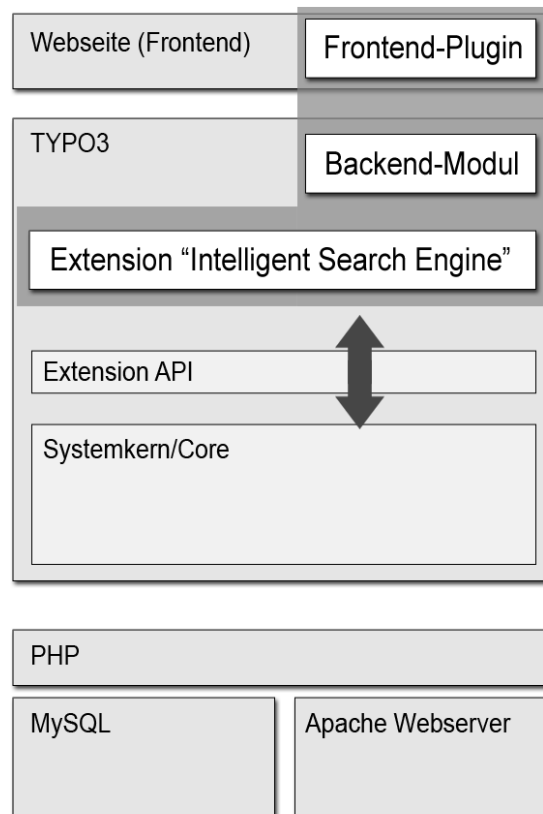


Abbildung 6: Architektur „Intelligent Search Engine“

5.2.2 Frontend-Plugin

Das Frontend-Plugin beinhaltet die Hauptfunktionalität der Extension. Es wird auf der jeweiligen Webseite über das TYPO3-Backend eingebunden und liefert ein Suchformular. Zusätzlich werden die Ergebnisse einer Suchanfrage vom Frontend-Plugin ausgegeben. Jedes Ergebnis bietet eine Bewertungsmöglichkeit. Die von der Benutzerin/vom Benutzer getätigten Bewertungen werden vom Frontend-Plugin entgegen genommen und verarbeitet.

5.2.3 Backend-Modul

Das Backend-Modul ist direkt über das TYPO3-Backend erreichbar. Es dient einerseits statistischen Zwecken, andererseits stellt es wichtige Funktionen für die Erweiterung zur Verfügung.

5.2.4 Datenbank

Für die Funktion der „Intelligent Search Engine“ werden 2 Datenbanktabellen benötigt.

5.2.4.1 Tabelle „tx_isearch_swords“

Die Tabelle enthält Einträge für alle Suchstrings, nach denen gesucht wurde. Die Tabelle „tx_isearch_swords“ beinhaltet folgende Felder:

Feldname	Beschreibung
swordId	Eindeutige „id“ des Suchstrings
sword	Suchstring der jeweiligen Anfrage
results	Anzahl der Resultate für die jeweilige Suchanfrage
timestamp	Timestamp zum Zeitpunkt der Suchanfrage
counter	Zähler, wie oft nach dem jeweiligen Suchstring gesucht wurde

Tabelle 4: Datenbanktabelle „tx_isearch_swords“

5.2.4.2 Tabelle „tx_isearch_results“

Die Tabelle enthält Einträge für alle Resultate, die bei den Suchanfragen gefunden wurden. Die Tabelle „tx_isearch_results“ beinhaltet folgende Felder:

Feldname	Beschreibung
resultsId	Eindeutige „id“ des Resultats
swordId	„id“ des Suchstrings aus der Tabelle „tx_isearch_swords“

title	Titel des jeweiligen Resultats
bodytext	Inhaltstext des jeweiligen Resultats
link	Link zur Seite des jeweiligen Resultats
counter	Zähler, wie oft der Suchstring im Resultat gefunden wurde
clicks	Zähler, wie oft auf das Resultat bereits geklickt wurde
evaluation	Stand der Bewertungen
startpoints	Startpunktezahl eines Resultats
points	Gesamtpunktestand des Resultats
expired	Feld dient zur Überprüfung, ob das Resultat gelöscht werden soll.

Tabelle 5: Datenbanktabelle „tx_isearch_results“

5.3 Umsetzung

Um die in Kapitel 4.1 getroffenen Anforderungen umzusetzen, wurden folgende Überlegungen getroffen und umgesetzt.

5.3.1 Was soll durchsucht werden?

Die Sucherweiterung durchsucht standardmäßig die Tabellen „tt_content“ sowie „pages“ der TYPO3-Datenbank auf Übereinstimmungen. Gesucht wird dabei in den Feldern „pages.title“, „tt_content.title“, „tt_content.bodytext“ und „tt_content.imagecaption“. Will man zusätzliche Tabellen durchsuchen, um installierte Erweiterungen in den Suchprozess einzubinden, kann dies per

Flexform-Konfiguration im TYPO3-Backend definiert werden. Zusätzlich wird auch eine Konfiguration per TypoScript angeboten.

5.3.2 Reihung der Ergebnisse

Um die Suchtreffer nach deren Relevanz zu reihen, wurden folgende Kriterien definiert, die bestmögliche Ergebnisse liefern sollen:

- **Bewertungen**
- **Anzahl der Klicks**
- **Anzahl der Treffer in einem Ergebnis**
- **Startpunkte**

Ermittelt wird der Rang eines Ergebnisses mit Hilfe eines Punktesystems. Es gibt für eine Bewertung, für einen Klick oder auch für einen Treffer in einem Ergebnis je einen Punkt. Eine negative Bewertung subtrahiert einen Punkt. Diese Punktestände werden für die Kriterien separat in der Datenbank gespeichert. Bewertungen und Klicks werden mit einer Gewichtung, die von der Benutzerin/vom Benutzer per Flexform-Konfiguration bzw. TypoScript definiert werden kann, multipliziert. Zusätzlich ist es möglich, dass ein Resultat so genannte „Startpunkte“ erhält. Näheres dazu finden Sie in Kapitel 5.3.2.4. Die Summe der Punktestände der verschiedenen Kriterien ergibt den Gesamtpunktestand eines Resultats. Dieser ist entscheidend für die Reihung des jeweiligen Ergebnisses.

5.3.2.1 Bewertungen

Das Herz der Erweiterung stellt die Möglichkeit der Resultats-Bewertung dar. Es soll ein jeder Suchtreffer evaluiert werden können. Dieser Vorgang soll schnell von statten gehen und nicht kompliziert sein. Die Benutzerin/Der Benutzer soll nicht den Eindruck haben, durch eine Bewertung aufgehalten zu werden.

Es gibt verschiedenste Möglichkeiten, ein Resultat bewerten zu lassen. Es gibt Systeme mit multiplen Bewertungsmöglichkeiten, die mit „1-x Sternen“, „1-x Punkten“, „Sehr gut bis Sehr schlecht“ usw. arbeiten. Die Benutzerin/Der Be-

nutzer steht bei solchen Systemen vor der Frage, wie viele Punkte/Sterne vergeben ich? Ist das Resultat gut, ist es mittelmäßig, ist es schlecht?

Ermöglicht dies einen raschen Bewertungsvorgang? Nein im Gegenteil, diese Gedankengänge verlangsamen den Entscheidungsprozess. Im Bezug auf ein Resultat einer Suchanfrage sind die Extreme interessant. Volle Anzahl an Punkten oder 0 Punkte, volle Anzahl an Sternen oder 0 Sterne, „Sehr gut“ oder „Sehr schlecht“, sprich einfach „gutes Resultat“ oder „schlechtes Resultat“. Dies ist eine Entscheidung, die die Benutzerin/der Benutzer schnell treffen kann. Entweder ein Resultat ist jenes, nach dem sie/er gesucht hat, oder eben nicht. Aus dieser Überlegung heraus werden als Bewertungsmöglichkeiten 2 Optionen angeboten: Entweder „Good“ oder „Bad“. Eine gute Bewertung addiert im entsprechenden Datenbankfeld einen Punkt, eine negative Bewertung hingegen subtrahiert einen Punkt. Der resultierende Bewertungs-Punktestand wird dann mit der definierten Gewichtung multipliziert. Es ist auch möglich, dass der Bewertungspunktestand negativ ist, wenn mehr negative als positive Bewertungen abgegeben wurden.

5.3.2.2 Anzahl der Klicks

Eine wichtige Kennzahl für die Reihung der Suchergebnisse stellt die Anzahl der Klicks auf ein bestimmtes Ergebnis dar. Ein Klick soll von der Gewichtung her mehr Wert sein als eine Bewertung. Er soll von der Logik her gleichbedeutend mit dem Finden des gesuchten Inhalts sein. Vom Prinzip her funktioniert die Ermittlung des Klick-Punktestandes identisch mit der des Bewertungspunktestandes. Ein Klick addiert einen Punkt im entsprechenden Datenbankfeld und wird später mit der definierten Gewichtung multipliziert.

5.3.2.3 Anzahl der Treffer in einem Ergebnis

Ein weiteres Kriterium, das bei der Reihung der Suchergebnisse berücksichtigt wird, ist die Anzahl der Treffer in einem Ergebnis. Es wird ermittelt, wie oft der gesuchte String in einem Ergebnis vorkommt. Diese Anzahl ist entscheidend für die Reihung der Suchergebnisse, wenn noch keine Bewertungen bzw. Klicks stattgefunden haben. Dieses Kriterium kann nicht per Gewichtung konfiguriert

werden. Es ist standardmäßig eine Gewichtung von 1 definiert, um die größere Wichtigkeit der Bewertungen und Klicks zu sichern.

5.3.2.4 Startpunkte

Das letzte Kriterium, das die Reihung der Suchergebnisse beeinflussen kann, sind so genannte „Startpunkte“. Wann werden an ein Resultat „Startpunkte“ vergeben? Am besten lässt sich dies mit einem Beispiel erläutern. Zum Suchwort „josie“ sind 5 Resultate gespeichert. Diese weisen bereits viele Bewertungen und Klicks auf, sprich sie verfügen über einen hohen Gesamtpunktestand. Wird nun ein neues Inhaltselement, bei dem der String „josie“ vorkommt, angelegt, wird dieses bei der nächsten Suche gefunden. Dieses neue Resultat würde allerdings auf Grund der fehlenden Bewertungen bzw. Klicks an das Ende der Suchtreffer gereiht werden. Dadurch würden neue Resultate von Beginn an benachteiligt werden. Um dies zu verhindern, werden dem Resultat „Startpunkte“ zugeteilt. Es wird dazu der Durchschnittsgesamtpunktestand aller Resultate betreffend des jeweiligen Suchstrings ermittelt. Der errechnete Durchschnitt wird dem Resultat als „Startpunktewert“ zugeteilt. Die „Startpunkte“ werden bei der Berechnung des Gesamtpunktestandes miteinbezogen. Dieses „Startpunkte-Konzept“ garantiert somit, dass neue Resultate im Mittelfeld der Resultatsreihung starten.

5.3.2.5 Beispiel für eine Gesamtpunkteberechnung

Gewichtungen:

Bewertungen: 5

Klicks: 10

Ein Resultat liefert folgende Werte:

Bewertungen: 5 Positive, 3 Negative

Klicks: 4

Treffer: 4

Berechnung:

Bewertungen: $5 - 3 = 2$ $2 * 5 = 10$ **Punkte**

Klicks: $4 * 10 = 40$ **Punkte**

Treffer: **4 Punkte**

Gesamtpunktestand: $10 + 40 + 4 = 54$ **Punkte**

„Startpunkte“ würden zusätzlich dem Gesamtpunktestand addiert werden.

5.3.3 Spracheinstellungen

Um eine mehrsprachige Sucherweiterung zu garantieren, wurden die entsprechenden „Locallang-Dateien“ erstellt. Dort wurde die Standardsprache Englisch sowie die Alternativsprache Deutsch angelegt und auch mit Inhalten befüllt. Eine Erweiterung der Sprachen ist daher ohne großen Aufwand durch Ergänzung in den jeweiligen Sprachdateien möglich.

5.3.4 Templates

Für das Frontend-Plugin wurde ein Template angelegt. Dieses Template kann entweder direkt verändert werden oder auch per Flexform-Konfiguration bzw. TypoScript durch Angabe eines alternativen Pfades ausgetauscht werden. Das Standard-Template ist sehr einfach aufgebaut. Es werden die entsprechenden Marker vom PHP-Script durch Inhalte ersetzt. Für das Aussehen ist ein externes Stylesheet zuständig.

```
1.      <!-- ###ISEARCH### begin -->
2.      ###SEARCH_FORM###
3.      <br/><br/>
4.      ###SEARCH_WORD###
5.      <br/><br/>
6.      ###PAGINATION###
7.      ###EVALUATION###
8.      ###RESULTS###
9.      ###PAGE_PREVIEW###
10.     <!-- ###ISEARCH### end -->
```

Listing 1: Template „Intelligent Search Engine“

Will man den Aufbau des Frontend-Plugins ändern, muss man nur die entsprechenden Marker anders positionieren.

5.3.5 Statistik

Für eine Statistik zu den Suchanfragen ist ein Backend-Modul zuständig. Es liefert eine Auflistung der einzelnen Suchanfragen sowie die Anzahl der dabei erzielten Resultate. Zusätzlich hat man die Möglichkeit, einzelne Suchbegriffe aus der Datenbank zu löschen.

ALLGEMEINE STATISTIK			
Anzahl der Suchbegriffe: 25			
Suchwort	Resultate	Anzahl	Löschen
josie	5	12	X
wirt	3	5	X
Hasselhof	1	4	X
zeit	3	3	X
dank	2	2	X

Abbildung 7: Statistik Suchwörter

Bei Klick auf das jeweilige Suchwort bekommt man nähere Informationen zu den einzelnen Resultaten.

ALLGEMEINE STATISTIK							
Resultate für den Suchbegriff: josie							
Zurück							
Rang	Titel	Link	Anzahl Klicks	Bewertung	Startpunkte	Punkte	
1	Neuer Eintrag	index.php?id=39	1	1	1	0	16
2	Allgemein	index.php?id=27	1	0	2	0	11
3	Links	index.php?id=21	1	0	0	7	7
4	Speisekarte	index.php?id=24	1	1	-1	0	6
5	Kontakt	index.php?id=22	1	0	0	0	1

Abbildung 8: Statistik Resultate

5.3.6 Usability

Es wurde bei der Erstellung der Erweiterung viel Wert auf deren Usability gelegt. Die Installation der Erweiterung erfolgt über das TYPO3-Standardmodul „Extension Manager“. Dieser Vorgang ist intuitiv und funktioniert rasch. (siehe auch Kapitel 3.1.7)

Ein wichtiger Punkt ist die Darstellung der Ergebnisse. Es ist entscheidend für die Erweiterung, dass die Benutzerinnen/Benutzer möglichst viele Bewertungen abgeben. Dies verbessert die Relevanz der erstgereihten Suchergebnisse und beschleunigt gleichzeitig den Suchprozess. Um jedoch ein Ergebnis bewerten zu können, müssen möglichst viele Informationen dazu vorhanden sein. Eine Listenansicht würde viele Ergebnisse auf einmal liefern. Dadurch hätte man das Problem, dass zu den einzelnen Resultaten nur beschränkt Informationen angezeigt werden könnten, ohne dass das Ganze unübersichtlich wird. Aus diesem Grund fiel die Wahl auf die Einzelansicht jedes Ergebnisses. Diese Darstellung ermöglicht die Anzeige von mehr Information. Zusätzlich wird die Aufmerksamkeit der Benutzerin/des Benutzers auf genau das derzeit angezeigte Resultat gelenkt. Dadurch kann sie/er sich auf dieses konzentrieren und entscheiden, wie sie/er es bewertet. Außerdem wird einer Bewertung zusätzliche Bedeutung verliehen, da der Rang in einer Einzelansicht wesentlich entscheidender ist, als vergleichsweise in einer Listenansicht.

Die Einzelansicht bietet die Möglichkeit, den Titel und den gesamten textlichen Inhalt des gefundenen Datensatzes anzuzeigen. Im Text wird das gefundene Suchwort hervorgehoben. Oft ist es jedoch trotz der Anzeige des ganzen Textes noch nicht wirklich möglich, eine Bewertung abzugeben. Es wäre gut, eine Art „Vorschau“ der jeweiligen Seite zu erhalten. Eine derartige Vorschau ist deshalb über das Lupensymbol neben dem Ergebnistitel möglich. Per Klick darauf öffnet sich die Zielseite in einem „DIV-Layer“, der per Klick wieder geschlossen werden kann.

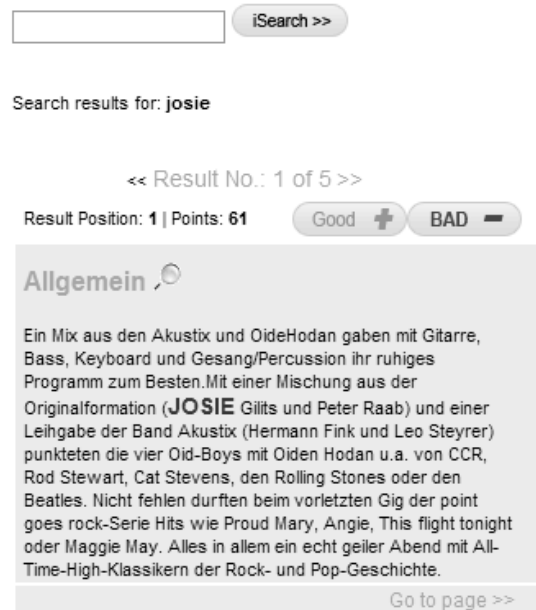


Abbildung 9: Einzelansicht Resultate

Die Navigation oberhalb des Resultats ermöglicht das Wechseln zum nächsten bzw. vorherigen Ergebnis. Darunter links befinden sich Informationen über den Rang des Ergebnisses sowie dessen Punktestand. Rechts hat man die Möglichkeit, das jeweilige Resultat zu bewerten. Nach einer Bewertung werden der Rang und auch der Punktestand neu berechnet.

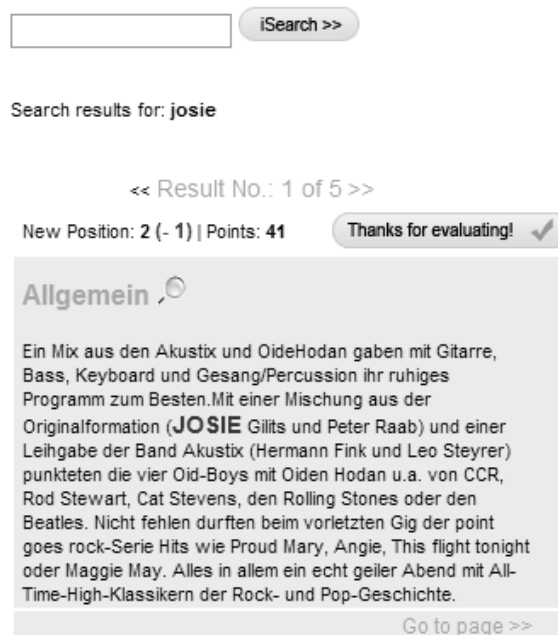


Abbildung 10: Ergebnis nach Bewertung

Wie in Abbildung 10 zu sehen ist, wurde nach der Bewertung des Resultats der Rang und der Punktestand neu berechnet. Ganz wichtig für die Usability ist jedoch die Tatsache, dass das Resultat betreffend Navigation trotzdem an derselben Position bleibt („Result No. 1 of 5“). Erst bei der nächsten Suche nach dem String wird die neue Reihenfolge ausgegeben. Hat man das gesuchte Ergebnis gefunden, kann man per Klick auf „Go to page“ auf die entsprechende Seite gelangen. Dem Resultat werden die Punkte für den Klick gutgeschrieben.

Um die Usability der Anwendung zusätzlich zu erhöhen, wurde auf AJAX gesetzt. Es wird das Browserfenster während des gesamten Such-, Navigier- bzw. Bewertungsvorganges kein einziges Mal neu geladen. Zusätzlich wird bei der Eingabe eines Zeichens in das Suchformular eine Autovervollständigung mit bereits gesuchten Begriffen und der Anzahl an gefundenen Resultaten angezeigt. Vorbild für diese Idee war „Google Suggest“ (<http://www.google.com/webhp?hl=EN>).



Abbildung 11: Autovervollständigung á la „Google Suggest“

5.3.7 Dokumentation

Um einer Benutzerin/einem Benutzer die Verwendung der Sucherweiterung so leicht wie möglich zu gestalten, wurde ein Handbuch erstellt. Dieses ist direkt in der Erweiterung enthalten und steht somit jederfrau/jedermann zur Verfügung.

5.4 Implementierung

Die Implementierung der Sucherweiterung brachte wichtige Schritte und Überlegungen mit sich. Diese werden im folgenden Abschnitt angesprochen.

5.4.1 Start mit Extension „Kickstarter“

Zu Beginn wurde mit Hilfe der Extension „Kickstarter“ die Grundstruktur der Erweiterung samt Struktur für Frontend-Plugin und Backend-Modul erstellt. Auch der Aufbau der Datenbanktabellen wurde definiert. Voraussetzung ist die Installation der Erweiterung „Kickstarter“ über das TYPO3-Backend. (siehe auch Kapitel 3.6).

The screenshot shows the 'Extension Manager' interface in TYPO3, specifically the 'KICKSTARTER WIZARD' for the 'Intelligent Search Engine' extension (i_search). The interface is divided into two main sections: a left sidebar with a list of configuration steps and a main area for entering details.

Left Sidebar (KICKSTARTER WIZARD):

- General info (selected)
- Intelligent Search Engine (with a trash icon)
- Setup languages (with a trash icon)
- [Click to Edit] (with a trash icon)
- New Database Tables (+)
- Extend existing Tables (+)
- Frontend Plugins (+)
- Intelligent Search (with a trash icon)
- Backend Modules (+)
- Search Statistic (with a trash icon)
- Integrate in existing Modules (+)
- Clickmenu items (+)
- Services (+)
- Static TypoScript code (+)
- TSconfig (+)

Main Area (General info):

Enter extension key:

General info
Enter general information about the extension here: Title, description, category, author...

Title:

Description:

Category:

State:

Dependencies (comma list of extkeys):

Author Name:

Author email:

At the bottom, there are buttons for 'Total form', 'View result', 'D/L as file', and a checkbox for 'Print WOP comments'.

Abbildung 12: Erstellung der Grundstruktur mit dem „Kickstarter“

Die erstellte Dateistruktur entspricht zum Großteil der standardmäßigen Struktur einer TYPO3-Erweiterung. Es wurden die Konfigurationsdateien („ext_emconf“,

„ext_localconf.php“), ein Icon für die Erweiterung („ext_icon.gif“), die „ext_tables.sql“ für den Datenbanktabellenimport sowie die Datei „local-lang_db.xml“ für die verschiedenen Sprachversionen angelegt. Zusätzlich wurde für das Frontend-Plugin der Ordner „pi1“ und für das Backend-Modul der Ordner „mod1“ mit den entsprechenden Dateien erstellt.

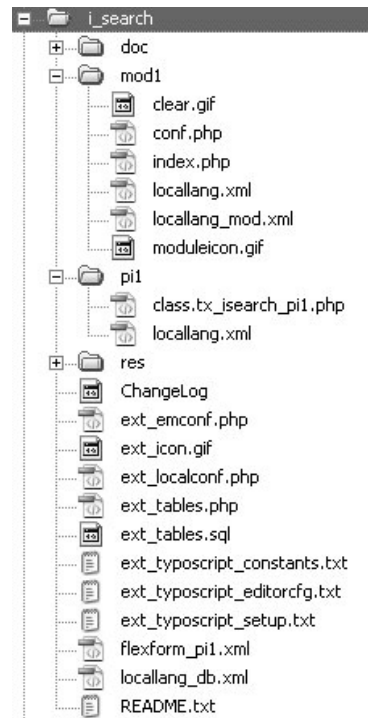


Abbildung 13: Dateistruktur der „Intelligent Search Engine“

Eine Abweichung von der Standard-Dateistruktur einer TYPO3-Erweiterung stellt die Datei „flexform_pi1.xml“ dar. Sie wird über die Datei „ext_tables.php“ eingebunden und beinhaltet eine Flexform „data structure“ (siehe auch Kapitel 3.4). In der Datei „flexform_pi1.xml“ wird die genaue Definition des Plugin-Konfigurationsformulars im Backend vorgenommen.

Eine weitere Abweichung zur Standard-Dateistruktur stellen die Dateien „ext_typoscript_constants.txt“, „ext_typoscript_editorcfg.txt“ und „ext_typoscript_setup.txt“ dar. In diesen Dateien befinden sich TypoScript-Anweisungen, die eine zusätzliche TypoScript-Konfigurationsmöglichkeit schaffen (siehe auch Kapitel 2.5).

5.4.2 Das Frontend-Plugin

Wie bereits erwähnt, wurde für das Frontend-Plugin der Ordner „pil“ angelegt. Dieser beinhaltet die Datei „class.tx_isearch_pil.php“, in der die Funktionalität des Frontend-Plugins untergebracht ist. Zusätzlich wurde eine Datei namens „locallang.xml“ erstellt, die die verschiedenen Sprachversionen enthält.

Das erste Problem, um das man sich bei einem Frontend-Plugin kümmern muss, ist dessen Konfiguration zu Beginn. Jedes Plugin wird bei dessen Einbindung in TYPO3 zuerst per Formular bzw. TypoScript konfiguriert.

5.4.2.1 Die Konfiguration per Flexform

Das Konfigurationsformular muss genau definiert werden. Welche Felder kommen darin vor, welchen Feldtypen besitzen diese usw.

The screenshot shows the configuration interface for the 'Intelligent Search' plugin. At the top, there is a dropdown menu labeled 'Erweiterung:' with 'Intelligent Search' selected. Below this is a section titled 'Plugin Optionen:' containing four tabs: 'Grundeinstellungen', 'Template/CSS', 'Gewichtung' (which is active), and 'Seitenvorschau'. Under the 'Gewichtung' tab, there are two input fields: 'Bewertungen' with the value '20' and 'Klicks' with the value '10'.

Abbildung 14: Das fertige Konfigurationsformular (Tab „Gewichtung“)

Für die Definition des Konfigurationsformulars wurde ein Flexform gewählt. Das Formular wird daher in einer XML „data structure“ definiert. Die „data structure“ für das in Abbildung 14 aktive Tab „Gewichtung“ ist in folgendem Listing zu sehen.

```

11.     <tx_i_search_s_weight>
12.         <ROOT>
13.             <TCEforms>
14.
15.                 <sheetTitle>LLL:EXT:i_search/locallang_db.xml:tt_content.pi
16.                 _flexform.s_weight</sheetTitle>
17.             </TCEforms>
18.             <type>array</type>
19.             <el>
20.                 <tx_i_search_evaluation>
21.                     <TCEforms>
```

```

22.         <label>LLL:EXT:i_search/locallang_db.xml:
23.         tt_content.pi_flexform.evaluation</label>
24.         <config>
25.             <type>input</type>
26.             <eval>num</eval>
27.             <size>10</size>
28.         </config>
29.     </TCEforms>
30. </tx_i_search_evaluation>
31. <tx_i_search_click>
32.     <TCEforms>
33.         <label>LLL:EXT:i_search/locallang_db.xml:
34.         tt_content.pi_flexform.click</label>
35.         <config>
36.             <type>input</type>
37.             <eval>num</eval>
38.             <size>10</size>
39.         </config>
40.     </TCEforms>
41. </tx_i_search_click>
42. </el>
43. </ROOT>
44. </tx_i_search_s_weight>

```

Listing 2: Auszug aus der Flexform „data structure“

Feldnamen werden dabei aber nicht direkt in die „data structure“ geschrieben, sondern über eine „Locallang-Referenzierung“. So wird z.B. in Listing 2 Zeile 14 der „Sheet Title“ per Referenz angegeben. In der Datei „locallang_db.xml“ findet man dazu folgende Einträge:

```

45.     <languageKey index="default" type="array">
46.     ...
47.     <label index="tt_content.pi_flexform.s_weight">Weight</label>
48.     ...
49.     </languageKey>
50.     <languageKey index="de" type="array">
51.     ...
52.     <label
53.         index="tt_content.pi_flexform.s_weight">Gewichtung</label>
54.     ...
55.     </languageKey>

```

Listing 3: Auszug „locallang_db.xml“

Wie in Listing 3 zu sehen, befindet sich in der Datei „locallang_db.xml“ je ein Eintrag für die „Standard-Sprache“ Englisch und je ein Eintrag für die Sprache „Deutsch“. Identifiziert wird der richtige Eintrag über den „label-index“, welcher in unserem Beispiel „tt_content.pi_flexform.s_weight“ ist und für die Tab-

Überschrift „Gewichtung“ bzw. „Weight“ steht. Gespeichert wird die Flexform „data structure“ im entsprechenden Datenbanktabellenfeld „pi_flexform“ in der Tabelle „tt_content“. Die im Flexform definierten Einstellungen werden im PHP-Script des Plugins über eine Funktion geladen und können somit im gesamten Frontend-Plugin verwendet werden.

5.4.2.2 Konfiguration per TypeScript

Zusätzlich zu der Konfiguration per Flexform ist auch eine TypeScript-Konfiguration möglich. Dazu wurden wie bereits in Kapitel 5.4.1 erwähnt die entsprechenden „txt-Dateien“ angelegt. Dort werden Felder definiert, auf die man später im TypeScript-Setup bzw. in den TypeScript-Constants zugreifen kann.

```
56.     plugin.tx_isearch_pi1 {
57.         enable = {$plugin.isearch.enable}
58.         expires = {$plugin.isearch.expires}
59.         search = {$plugin.isearch.search}
60.         submit = {$plugin.isearch.submit}
61.         alias = {$plugin.isearch.alias}
62.         tssite = {$plugin.isearch.tssite}
63.         startingpoint = {$plugin.isearch.startingpoint}
64.         recursive = {$plugin.isearch.recursive}
65.         tpl = {$plugin.isearch.tpl}
66.         style = {$plugin.isearch.style}
67.         evaluation = {$plugin.isearch.evaluation}
68.         click = {$plugin.isearch.click}
69.         width = {$plugin.isearch.width}
70.         height = {$plugin.isearch.height}
71.     }
```

Listing 4: TypeScript in der Datei „ext_typoscript_setup.txt“

Die in Listing 4 definierten Variablen werden dem PHP-Script des Frontend-Plugins in einem Array („\$conf“) übergeben und können somit im gesamten Plugin verwendet werden.

5.4.2.3 Die Datei „class.tx_isearch_pi1.php“

Diese Datei beinhaltet die gesamte Funktionalität des Frontend-Plugins. Die Klasse stellt eine Erweiterung der Klasse „tslib_pibase“ dar, der Standardklasse

für TYPO3 Frontend-Plugins. Die erste Funktion, die beim Aufruf des Frontend-Plugins aufgerufen wird, ist die „Main-Funktion“.

```
72.         function main($content,$conf)  {
73.
74.             //Loading TypoScript array into object variable
75.             $this->conf=$conf;
76.             $this->pi_setPiVarDefaults();
77.
78.             //Loading language labels
79.             $this->pi_loadLL();
80.
81.             // Configuring so caching is not expected.
82.             $this->pi_USER_INT_obj=1;
83.
84.             // path to extension
85.             $this->pathToExt = t3lib_extMgm::siteRelPath($this-
>extKey);
86.
87.             //prepare XAJAX
88.             $this->prepareXajax();
89.
90.             //include stylesheet
91.             $this->includeStylesheet();
92.
93.             //include javascript
94.             $this->includeJavaScript();
95.
96.             //init flexform variables
97.             $this->init();
98.
99.             //load template File
100.            $this->loadTemplate();
101.
102.            //fill marker with real content
103.            $content=$this->cObj->substituteMarkerArrayCached($this-
>templateFile,$this->marker);
104.
105.            //output content
106.            return $this->pi_wrapInBaseClass($content);
107.        }
```

Listing 5: „Main-Funktion“ der Datei „class.tx_isearch_pi1.php“

Der „Main-Funktion“ jedes Plugins werden vorhandene Inhaltselemente über das Array „\$content“ und vorhandene TypoScript-Konfigurationen über das Array „\$conf“ übergeben. Zu Beginn der Funktion wird die „Locallang-Datei“ geladen. Außerdem wird das Caching der Erweiterung deaktiviert.

Ein wichtiger Schritt stellt der Aufruf der XAJAX-Konfigurationsfunktion „\$this->prepareXajax()“ dar. Es wird die Klasse der benötigten XAJAX-Extension in-

kludiert und instanziiert. Zusätzlich werden folgende Funktionen in der „XAJAX-Instanz“ registriert.

- **Preview()**

Die „Preview-Funktion“ ist zuständig für die Autovervollständigung.

- **PreparePagination()**

Mit Hilfe der Funktion „PreparePagination()“ erfolgt die Ausgabe der Navigation.

- **Evaluation()**

Die Funktion „Evaluation()“ sorgt für die Verarbeitung der Bewertungen.

- **AddClick()**

In der Funktion „AddClick“ wird ein Klick auf ein Resultat verarbeitet.

- **startSearch()**

Eine Suchanfrage wird von der Funktion „startSearch()“ abgearbeitet.

Abschließend wird XAJAX mit der Funktion „\$this->xajax->processRequests()“ veranlasst, auf alle „AJAX-Requests“ zu reagieren.

Danach werden in der „Main-Funktion“ das Stylesheet und auch das nötige JavaScript per Funktion eingebunden. Nächster Schritt ist die Initialisierung der Flexform-Variablen. Ist dies erledigt folgt noch das Laden der „Template-Datei“ über die Funktion „\$this->loadTemplate()“. In dieser Funktion wird zusätzlich zum Laden des Templates auch das entsprechende „Marker-Array“ mit den „Standard-Inhalten“ gefüllt. Im letzten Schritt in der „Main-Funktion“ erfolgt das Ersetzen der Template-Marker durch die Inhalte des vorher genannten „Marker-Array“. Der resultierende „HTML-Code“ wird in die Variable „\$this->content“ gespeichert und schließlich mittels „\$this->pi_wrapInBaseClass(\$content)“ ausgegeben.

Die „Main-Funktion“ wird beim Aufruf des Frontend-Plugins nur einmal ausgeführt. Alle anderen Anfragen werden von den „XAJAX-Funktionen“ verarbeitet, sprich die Seite wird auch nicht mehr neu geladen.

5.4.2.4 Der Suchvorgang

Es wird ein Suchbegriff in das Suchformular eingegeben. Bei jeder einzelnen Zeicheneingabe wird ein „AJAX-Request“ an die „Preview-Funktion“ gesendet. Diese Funktion liefert die „Autovervollständigung“ (siehe auch Kapitel 5.4.2.5). Wird eine Eingabe mittels Enter-Taste oder per Klick auf den „isearch-Button“ bestätigt, wird ein „AJAX-Request“ an die Funktion „start_search()“ gesendet.

Erster Schritt in der Funktion „start_search()“ ist eine Überprüfung, ob der übermittelte Suchstring bereits in der Datenbanktabelle „tx_isearch_swords“ gespeichert ist. Ist der Suchstring noch nicht in der Datenbank gespeichert, wird das entsprechende „SQL-Statement“ unter Berücksichtigung der Flexform- bzw. TypoScript-Konfiguration zusammengesetzt und eine neue Suchanfrage gestartet. Wird mindestens ein Resultat gefunden, wird der gesuchte String in die Datenbanktabelle „tx_isearch_swords“ und die gefundenen Resultate in die Datenbanktabelle „tx_isearch_results“ geschrieben. Abschließend werden die Resultate ausgegeben.

Ist der Suchstring bereits in der Datenbank, erfolgt eine Überprüfung, ob der gesuchte String bereits „abgelaufen“ ist. Dazu eine Erläuterung. Man hat per Flexform bzw. TypoScript die Möglichkeit, als „Ablaufzeit“ eine Zeitspanne in Stunden zu definieren. Am besten kann man dies mit einem Beispiel erläutern. Für das Suchwort „josie“, das bereits in der Datenbank gespeichert ist, wurde eine Zeitspanne von 48 Stunden per Flexform bzw. TypoScript als „Ablaufzeit“ definiert. Erfolgt ein erneuter Aufruf des Suchstrings „josie“ innerhalb von 48 Stunden, werden sofort die in der Tabelle „tx_isearch_results“ gespeicherten Resultate ausgegeben. Erfolgt jedoch ein Aufruf nach diesen 48 Stunden, wird neu in den TYPO3-Inhaltsdatenbanktabellen gesucht. Dieses „Ablaufzeit-Konzept“ ermöglicht einen rascheren Suchvorgang, da man sich die komplexere Abfrage in den TYPO3-Inhaltstabellen ersparen kann. Ist der Suchstring noch nicht „abgelaufen“, werden demnach sofort die in der Datenbank gespeicherten Resultate ausgegeben.

Ist der Suchstring hingegen „abgelaufen“, müssen einige Dinge beachtet werden. Als erstes wird eine neue Suche in den TYPO3-Inhaltstabellen gestartet. Wird

nichts gefunden, werden alle Einträge aus den Tabellen „tx_isearch_swords“ bzw. „tx_isearch_results“ gelöscht. Wird etwas gefunden, wird es problematisch. Es ist nämlich nicht möglich zu sagen, es werden alle alten Datensätze aus den Tabellen gelöscht und die neu gefundenen in die Datenbank geschrieben, da man dadurch alle Bewertungen und Klicks verlieren würde. Es muss eine andere Lösung gefunden werden.

Eine neue Suche in den TYPO3-Inhaltstabellen bedeutet nicht, dass die zuvor gefundenen Einträge nicht mehr vorhanden sind. Es kann sein, dass zusätzliche Einträge gefunden werden oder auch, dass es einige Einträge nicht mehr gibt. Es kann aber auch vorkommen, dass ein Eintrag zwar noch existiert, es sich aber etwas in dessen Inhalt geändert hat. Ist dies der Fall, muss man überlegen, ob dann eine bereits gespeicherte Bewertung bzw. ein bereits gespeicherter Klick noch gültig ist. Dies wäre mit Vorsicht zu genießen, denn diese Bewertungen bzw. Klicks galten dem alten Suchtreffer, der sich vom Inhalt her vom neuen Treffer unterscheidet.

Aus diesen Überlegungen wurde folgende Vorgehensweise umgesetzt. Es wird eine neue Suche in den TYPO3-Inhaltstabellen durchgeführt. Danach wird jedes bereits gespeicherte „alte“ Resultat mit den „neuen“ Resultaten verglichen. Ist ein „altes“ Resultat identisch wieder in der neuen Suche gefunden worden, bleibt dieses und dessen Bewertungen in der Datenbank erhalten. Alle „alten“ Resultate, die nicht mehr mit exakt identischem Inhalt in der neuen Suche gefunden worden sind, werden aus der Datenbank gelöscht. Die neu gefundenen Suchtreffer, die nicht bereits in den Datenbanktabellen vorhanden waren, werden neu angelegt. Außerdem werden diesen neuen Resultaten die in Kapitel 5.3.2.4 erwähnten „Startpunkte“ zugeteilt. Ist diese Überprüfung und Eintragung in die Datenbank abgeschlossen, werden die Ergebnisse ausgegeben.

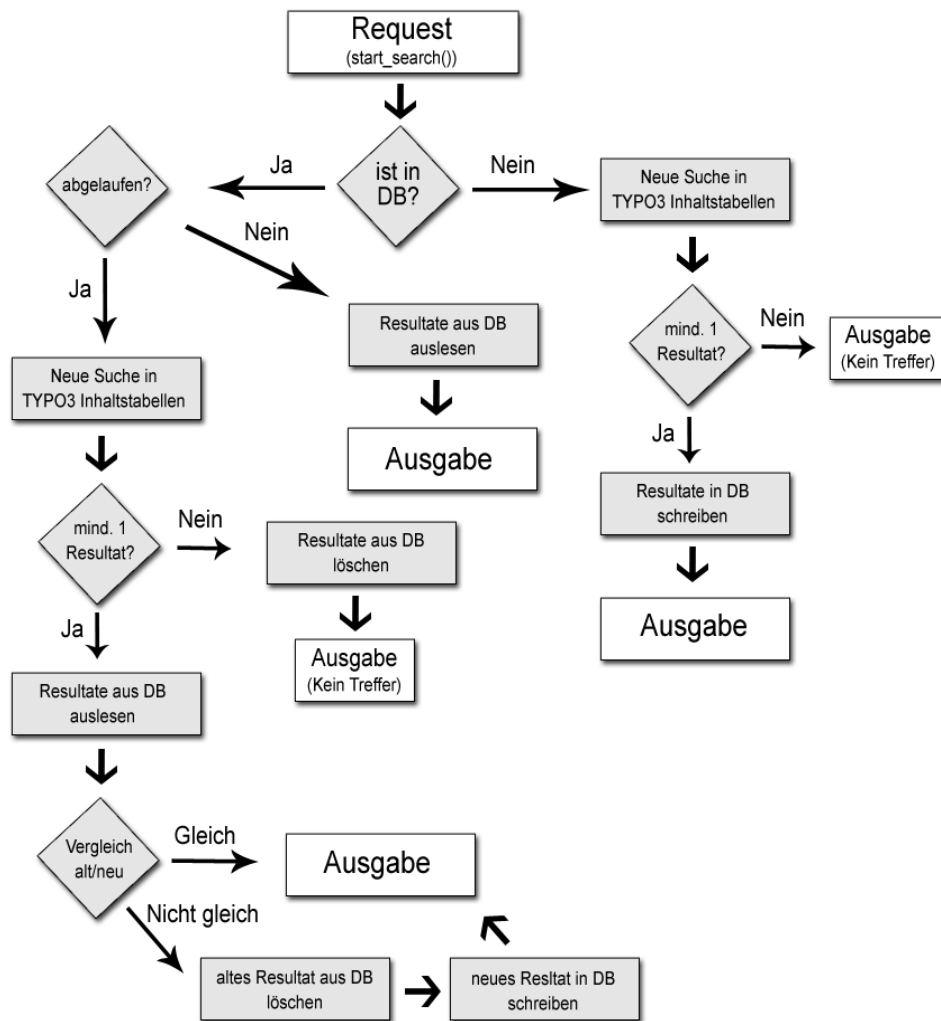


Abbildung 15: Flussdiagramm Suchvorgang

5.4.2.5 Die Autovervollständigung

Es wird bei Eintippen jedes Zeichens in das Suchformular ein „AJAX-Request“ an die Funktion „Preview()“ gesendet. Der übermittelte Suchstring wird mit den in der Tabelle „tx_isearch_swords“ gelisteten Suchstrings abgeglichen. Bei Übereinstimmungen werden diese Suchstrings und die Anzahl der jeweiligen Resultate aus der Datenbank ausgelesen. Werden Datensätze gefunden, wird ein „DIV-Layer“ unterhalb des Suchformulars mit den Einträgen gefüllt und eingeblendet.

5.4.2.6 Der Bewertungsvorgang

Die Bewertung erfolgt wiederum über einen „AJAX-Request“. Egal ob Bewertung „Good“ oder Bewertung „Bad“ gewählt wird, es wird die Funktion „evaluation()“ aufgerufen. Zu Beginn wird das Datenbanktabellenfeld „evaluation“ des jeweiligen Resultats ausgelesen. Handelt es sich um eine positive Bewertung, wird der Wert des Feldes um 1 Zähler erhöht. Handelt es sich hingegen um eine negative Bewertung wird der Wert des Feldes um 1 Zähler verringert. Das Datenbanktabellenfeld muss nun nur noch mit dem neuen Wert überschrieben werden.

Im nächsten Schritt muss der Gesamtpunktestand des Resultats neu berechnet werden. Dazu werden einige Komponenten benötigt. Einerseits müssen die Datenbanktabellenfelder „evaluation“, „clicks“, „counter“ und „startpoints“ ausgelesen werden. Andererseits werden für die Berechnung die Wertigkeiten eines Klicks und einer Bewertung benötigt. Diese Werte werden entweder aus dem Flexform-Array oder aus dem TypoScript-Array, die in der „Main-Funktion“ der Datei „class.tx_isearch_pi1.php“ initialisiert wurden, ausgelesen. Sind diese Komponenten vorhanden, kann der neue Punktestand berechnet werden. Schlussendlich wird das Datenbanktabellenfeld „points“ der Tabelle „tx_isearch_results“ mit dem neuen Punktestand überschrieben.

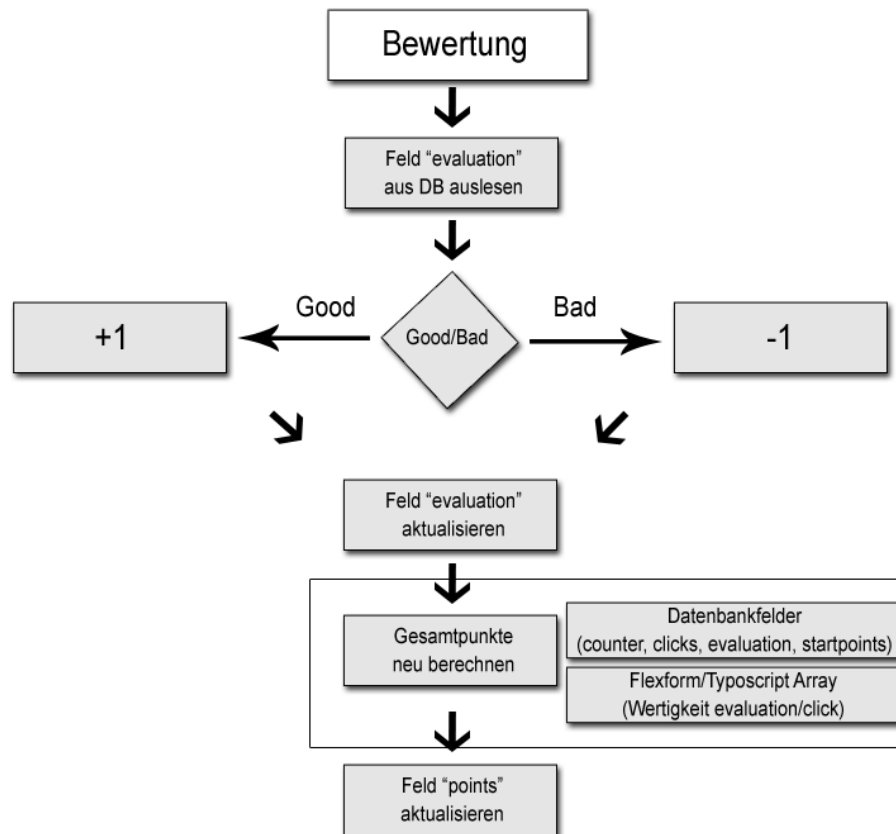


Abbildung 16: Flussdiagramm Bewertungsvorgang

5.4.2.7 Vorgang bei Klick auf ein Resultat

Der Vorgang bei Klick auf ein bestimmtes Resultat verläuft identisch zum Bewertungsvorgang in Kapitel 5.4.2.6. Einziger Unterschied ist, dass nicht das Datenbanktabellenfeld „evaluation“ sondern das Feld „clicks“ ausgelesen, um 1 erhöht und wieder überschrieben wird. Danach erfolgt die Berechnung des neuen Gesamtpunktestandes.

5.4.2.8 Der Reihungsvorgang

Ausschlaggebend für die Reihung der Suchtreffer ist der Gesamtpunktestand, der im Datenbanktabellenfeld „points“ der Tabelle „tx_isearch_results“ gespeichert ist. Wenn eine Benutzerin/ein Benutzer nach einem bestimmten String sucht, bekommt sie/er die Ergebnisse sortiert nach deren Punkteständen zurückgelie-

fert. Wichtig bei der Navigation durch diese Ergebnisse ist, dass sich die Positionen der Resultate nicht ändern. Dies wäre der Fall, wenn die Benutzerin/der Benutzer Resultate bewertet, und sich dadurch deren Punktestände ändern würden. Daher wird die bei der Suche zurück gelieferte Ergebnisreihung in einer Session gespeichert. Diese in der Session gespeicherte Reihung wird durch etwaige Bewertungen nicht beeinflusst. Erst wenn die Benutzerin/der Benutzer eine neue Suche nach demselben String startet, wird die neue Reihenfolge ausgegeben und die Session mit der neuen Reihung überschrieben.

5.4.3 Das Backend-Modul

Für das Backend-Modul wurde von der Extension „Kickstarter“ der Ordner „mod1“ angelegt. Darin befinden sich die Standarddateien eines TYPO3 Backend-Moduls. Die Datei „conf.php“ beinhaltet die Grundkonfiguration, „moduleicon.gif“ stellt das Extension-Icon für das TYPO3-Backend dar. In den Sprachdateien „locallang.xml“ bzw. „locallang_mod.xml“ befinden sich die textlichen Inhalte. Die Hauptfunktionalität findet man in der Datei „index.php“.

Bei der Erstellung eines Moduls muss man sich als erstes die Frage stellen, wo man das Modul in der Modulleiste einordnen möchte. Da es bei der Erweiterung „Intelligent Search Engine“ hauptsächlich um Inhalte der jeweiligen Webseite geht, wurde die Einordnung in das Hauptmodul „Web“ gewählt (siehe auch Kapitel 2.4).



Abbildung 17: Eingliederung in das Hauptmodul „Web“

5.4.3.1 Die Datei „index.php“

Diese Datei beinhaltet die gesamte Funktionalität des Backend-Moduls. Sie stellt eine Erweiterung der Klasse „t3lib_SCbase“ dar, der Standardklasse für TYPO3 Backend-Module.

Erste Funktion, die in der Datei aufgerufen wird, ist wie schon beim Frontend-Plugin die „Main-Funktion“. Dort werden benötigte globale Variablen initialisiert sowie Flexform- und TypoScript-Konfigurationen geladen. Außerdem wird der Dokumententyp festgelegt. Dazu gibt es vordefinierte Vorlagen, die von TYPO3 angeboten werden. Diese Vorlagen stellen Layout-Funktionen zur Verfügung, die für ein TYPO3-konformes Erscheinungsbild im Backend sorgen. Anschließend wird die Funktion „moduleContent()“ aufgerufen, die für das Rendern des Inhalts zuständig ist. Das Backend-Modul teilt sich in 2 Bereiche. Bereich 1 liefert eine allgemeine Statistik zum Frontend-Plugin (siehe auch Kapitel 5.3.5).

Folgende Funktionen werden dafür benötigt:

- **getWords()**
„getWords()“ liefert eine Auflistung aller angefragter Suchstrings mit Anzahl der Resultate und Anzahl der Klicks.
- **deleteWord()**
Die Funktion „deleteWord()“ löscht einen Eintrag aus der Tabelle „tx_isearch_swords“ sowie die dazugehörenden Resultate aus der Tabelle „tx_isearch_results“.
- **getResults()**
Durch Klick auf ein Suchwort werden mit Hilfe der Funktion „getResults()“ die dazugehörenden Resultate mit Angabe der Punktestände sowie Titel und Link des Resultats angezeigt.

Bereich 2 des Backend-Moduls beinhaltet wichtige Funktionen zur Steuerung des Frontend-Plugins.



Abbildung 18: Bereich 2: „Löschen und Aktualisieren“

Folgende Funktionen werden in diesem Bereich benötigt:

- **deleteWords()**

Diese Funktion ermöglicht das Löschen aller Einträge in den beiden Datenbanktabellen „tx_isearch_swords“ und „tx_isearch_results“.

- **recalc()**

„recalc()“ ist zuständig für die Neuberechnung der Gesamtpunktestände aller Resultate.

- **reset_evaluations()**

Mit Hilfe dieser Funktion wird das Datenbanktabellenfeld „evaluation“ aller Resultate auf 0 zurückgesetzt.

- **reset_clicks()**

„reset_clicks()“ setzt das Datenbanktabellenfeld „clicks“ aller Resultate auf 0 zurückgesetzt.

- **reset_startpoints()**

Die Funktion setzt das Datenbanktabellenfeld „startpoints“ aller Resultate auf 0 zurückgesetzt.

Zur Funktion „recalc()“ ist eine nähere Erläuterung nötig. Wann ist eine manuelle Neuberechnung der Gesamtpunktestände eigentlich nötig? Dies ist der Fall, wenn man im Nachhinein die Gewichtungen der Bewertungen und Klicks ändert. Eine Änderung der Konfiguration bedeutet nämlich nicht, dass dadurch auch die Gesamtpunktestände in der Datenbank geändert werden. Dort bleiben alle Resultate mit den alten Punkteständen gespeichert. Bei einer Suche nach einem bereits in der Datenbank gespeicherten String würden also alle Resultate noch mit den alten Punkteständen ausgegeben werden. Erst wenn eine Bewertung zu einem bestimmten Resultat abgegeben wird, wird für dieses Resultat der Punktestand neu berechnet, sprich die neuen Gewichtungen fließen mit ein. Das heißt, es müsste jedes Resultat einmal bewertet werden, um den mit den neuen Gewichtungen berechneten Gesamtpunktestand aufzuweisen. Aus diesem Grund gibt es die Funktion „recalc()“. Werden Änderungen an den Gewichtungen vorgenommen, sollte diese Funktion aufgerufen werden. Mit Hilfe der Funktion werden alle Gesamtpunktestände auf einmal neu berechnet.

Der von den verschiedenen Funktionen innerhalb von „moduleContent()“ zurück gelieferte HTML-Code wird in der Variable „\$content“ gespeichert. Nach Ausführung der „Main-Funktion“ wird der Inhalt der Variable „\$content“ von der Funktion „printContent()“ ausgegeben.

5.4.3.2 Flexform- bzw. TypoScript-Konfiguration

Um über das Backend-Modul die Neuberechnung der Gesamtpunktestände durchzuführen, müssen die definierten Flexform- bzw. TypoScript-Konfigurationen zur Verfügung stehen. Im Frontend-Plugin ist dies standardmäßig der Fall. Die TypoScript-Konfiguration wird der „Main-Funktion“ direkt übergeben, die Flexform-Konfiguration steht im globalen Frontend-Objekt „\$TSFE“ zur Verfügung (siehe auch Kapitel 3.3.1.1). So einfach funktioniert dies im Backend-Modul leider nicht. Es ist ein kleiner „Workaround“ notwendig, um auf die benötigten Werte zugreifen zu können.

Zuerst müssen die Flexform-Werte geladen werden. Diese können aus der Datenbanktabelle „tt_content“ aus dem Feld „pi_flexform“ in XML-Form ausgelesen werden.

```

108.    //Function to load Flexform vars
109.    function getFlexFormVars(){
110.
111.        //Query to read pi_flexform field from tt_content table
112.        $select_fields="pi_flexform";
113.        $from_table=$this->contenttable;
114.        $where_clause="list_type LIKE '".$this->plugin.'" AND
115.        hidden LIKE 0";
116.
117.        $res2=$GLOBALS['TYPO3_DB']-
118.        >exec_SELECTquery($select_fields,$from_table,
119.        $where_clause);
120.
121.        while($row2 = $GLOBALS['TYPO3_DB']-
122.        >sql_fetch_assoc($res2)) {
123.            $cfgArr = t3lib_div::xml2array($row2['pi_flexform']);
124.
125.        }
126.
127.        //store evaluation and click weight from flexform
128.        $this->evaluation=
129.        $cfgArr['data']['tx_i_search_s_weight']['lDEF']
130.        ['tx_i_search_evaluation']['vDEF'];
131.        $this->click=
132.        $cfgArr['data']['tx_i_search_s_weight']
133.        ['lDEF']['tx_i_search_click']['vDEF'];
134.        $this->pageUid=
135.        $cfgArr['data']['tx_i_search_s_general']
136.        ['lDEF']['tx_i_search_tssite']['vDEF'];
137.
138.    }

```

Listing 6: Flexform-Konfiguration laden

Die in XML-Form vorhandene Flexform-Konfiguration muss dann nur noch per TYPO3-Funktion „t3lib_div::xml2array()“ in ein Array konvertiert werden. Im Array „\$cfgArr“ stehen nun alle Flexform-Variablen zur Verfügung.

Nächster Schritt ist das Laden der TypoScript-Konfiguration. Dazu wird die „Seiten-id“ des „Main-Templates“ benötigt, da dieses die TypoScript-Konfiguration beinhaltet. Diese „Seiten-id“ muss in der Flexform-Konfiguration definiert werden, um im Backend-Modul auf die Typoscript-Werte zugreifen zu können. Wurde die „Seiten-id“ definiert, kann diese aus dem zuvor gefüllten Array „\$cfgArr“ ausgelesen werden. Diese „id“ wird dann an die Funktion „loadTS()“ übergeben, um die TypoScript-Konfiguration zu laden.

```
139.         function loadTS($pageUid) {
140.             $sysPageObj =
141.                 t3lib_div::makeInstance('t3lib_pageSelect');
142.             $rootLine = $sysPageObj->getRootLine($pageUid);
143.             $TSObj =
144.                 t3lib_div::makeInstance('t3lib_tsparser_ext');
145.             $TSObj->tt_track = 0;
146.             $TSObj->init();
147.             $TSObj->runThroughTemplates($rootLine);
148.             $TSObj->generateConfig();
149.             $this->conf = $TSObj-
150.                 >setup['plugin.']['tx_isearch_pi1.'];
151.         }
```

Listing 7: TypoScript-Konfiguration laden

Durch diesen „Workaround“ stehen auch im Backend-Modul alle per Flexform bzw. TypoScript definierten Werte zur Verfügung.

5.5 Veröffentlichung der Extension

Nach Fertigstellung der Erweiterung „Intelligent Search Engine“ wurde die Veröffentlichung im TYPO3 Online-Extension-Repository vorbereitet und durchgeführt. Hauptaugenmerk in der Vorbereitung lag in der Erstellung eines Handbuchs, dass den Benutzerinnen/Benutzern der Erweiterung die Konfiguration erleichtern soll. Der Veröffentlichungsvorgang selbst wird über das Modul „Extension Manager“ (siehe auch Kapitel 3.1.7) im TYPO3-Backend durchgeführt. Alle im jeweiligen Extension-Ordner liegenden Dateien werden per Klick zum TYPO3-Repository transferiert. Befindet sich im Ordner „doc“ eine Datei mit dem Namen „manual.sxw“ – das Handbuch – wird dieses zusätzlich für eine Online-Vorschau in „HTML-Code“ konvertiert, um dann auf der offiziellen TYPO3-Webseite <http://www.typo3.org> zur Verfügung zu stehen.

i Intelligent Search Engine (i_search)			Beta
Author	Category	Updated	
Mathias Moser	plugin	27.07.2007	
Version	Manual	Upload comment	
0.0.2	Read online		
Downloads (total / this version)	Rating		
38 / 38	none		
Description			
Search Engine which includes the User in the result sort process.			
Dependencies		Version history	
depends on cms			
depends on xajax			
Reverse dependencies			
The following extension versions depend on this extension:			
Download extension			
Download compressed extension .T3X file			
Filename	Size	View	Date
ChangeLog	108	View	21.06.2007
README.txt	80	View	21.06.2007
ext_icon.gif	129	View	04.07.2007
ext_localconf.php	371	View	21.06.2007
ext_tables.php	655	View	19.07.2007
ext_tables.sql	0.9 K	View	04.07.2007
ext_typoscript_constants.txt	1.9 K	View	24.07.2007
ext_typoscript_editorcfg.txt	0.9 K	View	21.06.2007
ext_typoscript_setup.txt	548	View	23.07.2007
flexform_pi1.xml	6.2 K	View	23.07.2007
locallang_db.xml	2.7 K	View	23.07.2007
doc/manual.sxw	25 K		24.07.2007
doc/wizard_form.dat	2.6 K		21.06.2007
doc/wizard_form.html	29 K		21.06.2007
mod1/Thumbs.db	4.0 K		19.07.2007
mod1/clear.gif	46	View	21.06.2007
mod1/conf.php	389	View	19.07.2007
mod1/index.php	23 K	View	24.07.2007
mod1/locallang.xml	3.2 K	View	24.07.2007
mod1/locallang_mod.xml	849	View	21.06.2007
mod1/moduleicon.gif	308	View	19.07.2007
pi1/class.tx_isearch_pi1.php	51 K	View	27.07.2007
pi1/locallang.xml	0.9 K	View	04.07.2007
res/img/Thumbs.db	12.5 K		05.07.2007
res/img/eval_bad.gif	1.0 K	View	03.07.2007
res/img/eval_good.gif	1.0 K	View	03.07.2007
res/img/eval_thanks.gif	1.9 K	View	03.07.2007
res/img/loading.gif	1.4 K	View	05.07.2007
res/img/page_preview.gif	616	View	04.07.2007
res/img/search_button.gif	1.0 K	View	03.07.2007
res/js/search.js	2.9 K		15.07.2007
res/styles/styles.css	2.1 K	View	15.07.2007
res/tmpl/example.tmpl	207		15.07.2007

Abbildung 19: „Intelligent Search Engine“ im TYPO3 Extension-Repository

Wie in Abbildung 19 zu sehen, wurde die Erweiterung am 27. Juli 2007 mit dem Entwicklungsstatus „beta“ veröffentlicht. Die Erweiterung ist somit über das TYPO3 Online-Extension-Repository frei zugänglich. Die Dateien der Erweiterung können entweder einzeln oder gesamt als komprimiertes „t3x-Paket“ heruntergeladen werden.

6 Fazit und Ausblick

Nachdem die Erweiterung bereits mit dem Entwicklungsstatus „beta“ veröffentlicht wurde, kann ein erstes Resümee gezogen werden. In erster Linie kann man sagen, dass die „Intelligent Search Engine“ alle gestellten Anforderungen erfüllt. Es ist eine intuitive, benutzerfreundliche Erweiterung entstanden, die die Benutzerin/den Benutzer in den Reihungsprozess der Resultate einbezieht und somit die Relevanz der erstgereihten Ergebnisse verbessert.

Die Erweiterung

- ermöglicht das Suchen in den TYPO3-Standardinhaltselementen sowie in beliebigen Erweiterungen.
- ermöglicht der Benutzerin/dem Benutzer die Bewertung jedes Suchtreffers.
- ist auf Mehrsprachigkeit ausgerichtet.
- basiert auf externen Designvorlagen.
- liefert eine Statistik im TYPO3-Backend.
- ist leicht zu installieren sowie intuitiv in der Bedienung.
- wird unterstützt durch ein mitgeliefertes Handbuch.
- ist seit 27.Juli 2007 im offiziellen TYPO3-Extension-Repository mit dem Entwicklungsstatus „beta“ veröffentlicht.

Die „Intelligent Search Engine“ hebt sich von anderen TYPO3-Sucherweiterungen ab. Keine andere Erweiterung beinhaltet ein integriertes Bewertungssystem. Seit Veröffentlichung der Erweiterung sind bereits einige Wochen vergangen. Das erste Feedback der TYPO3-Community fällt sehr positiv aus. Rund 40 Downloads in den ersten Tagen spiegeln das Interesse an der Erweiterung wider. In nächster Zeit werden Verbesserungen und Weiterentwick-

lungen nötig sein, um die Sucherweiterung in der TYPO3-Welt zu etablieren. Dies ist auf Grund des Entwicklungsstatus „beta“ auch vorgesehen. Die Entwicklung der Erweiterung soll laufend forciert und weiterverfolgt werden. Verbesserungsvorschläge der Community sollen aufgenommen und bei Bedarf umgesetzt werden.

Großes Ziel für die nächsten 12 Monate ist der Schritt zum Entwicklungsstatus „stable“. Zuerst muss man allerdings abwarten, wie sich die Erweiterung in der Praxis schlägt. Hat sie sich bewährt, kann der Schritt zum Entwicklungsstatus „stable“ vollzogen werden.

Um ein fertiges und erprobtes Softwareprodukt auf den Markt zu bringen, sind viel Entwicklungszeit sowie ein langer Entwicklungsprozess notwendig. Die „Intelligent Search Engine“ steht noch am Anfang dieses Entwicklungsprozesses. Auf Grund des innovativen Konzeptes steckt aber genug Potential in der Erweiterung, um die Entwicklung weiter zu verfolgen.

Literaturverzeichnis

Bücher/Handbücher:

- [1] **Stöckl, Bongers** (2006): Einstieg in Typo3 4.0. Galileo Computing
- [2] **Laborenz, Wendt, Ertel, Dussoye, Hinz** (2006): Typo3 4.0. Das Handbuch für Entwickler. Galileo Computing
- [3] **Meyer** (2006): Praxiswissen TYPO3. O'Reilly
- [4] **Skårhøj** (2007): Manual: Indexed Search Engine.
- [5] **Schmid** (2005): Manual: [AN] Search It!
- [6] **Haim** (2002): Manual: Ajax Searching Engine.

Weblinks:

- [7] <http://typo3.com/Geschichte.1268.0.html?&L=2>, 01.08.2007
- [8] <http://typo3.org/extensions/what-are-reviews/>, 01.08.2007
- [9] <http://typo3.org/documentation/document-library/extension-manuals/xajax/0.2.4/view/1/1/>, 01.08.2007
- [10] <http://www.teialehrbuch.de/Kostenlose-Kurse/AJAX/20801-Asynchroner-Seitenaufbau.html>, 01.08.2007

Abbildungsverzeichnis

Abbildung 1: Offizielles TYPO3-Logo	9
Abbildung 2: Systemaufbau von TYPO3 (vgl. [2], S.31)	11
Abbildung 3: Das TYPO3-Backend	14
Abbildung 4: Einbindung von HTML-Designvorlagen in TypoScript	16
Abbildung 5: TYPO3-Modul „Extension Manager“	26
Abbildung 6: Architektur „Intelligent Search Engine“	48
Abbildung 7: Statistik Suchwörter.....	55
Abbildung 8: Statistik Resultate	55
Abbildung 9: Einzelansicht Resultate	57
Abbildung 10: Ergebnis nach Bewertung.....	57
Abbildung 11: Autovervollständigung á la „Google Suggest“	58
Abbildung 12: Erstellung der Grundstruktur mit dem „Kickstarter“	59
Abbildung 13: Dateistruktur der „Intelligent Search Engine“	60
Abbildung 14: Das fertige Konfigurationsformular (Tab „Gewichtung“)	61
Abbildung 15: Flussdiagramm Suchvorgang.....	68
Abbildung 16: Flussdiagramm Bewertungsvorgang.....	70
Abbildung 17: Eingliederung in das Hauptmodul „Web“	71
Abbildung 18: Bereich 2: „Löschen und Aktualisieren“	73
Abbildung 19: „Intelligent Search Engine“ im TYPO3 Extension-Repository	77

Tabellenverzeichnis

Tabelle 1: Wichtige Tabellen in der TYPO3-Datenbank (vgl. [2], S.35)	12
Tabelle 2: Auszug aus der TYPO3-Verzeichnisstruktur (vgl. [2], S. 37)	13
Tabelle 3: Objekte im „\$TSFE“ (vgl. [2], S. 579)	29
Tabelle 4: Datenbanktabelle „tx_isearch_swords“	49
Tabelle 5: Datenbanktabelle „tx_isearch_results“	50

Listings

Listing 1: Template „Intelligent Search Engine“	54
Listing 2: Auszug aus der Flexform „data structure“	62
Listing 3: Auszug „locallang_db.xml“	62
Listing 4: TypoScript in der Datei „ext_typoscript_setup.txt“	63
Listing 5: „Main-Funktion“ der Datei „class.tx_isearch_pi1.php“	64
Listing 6: Flexform-Konfiguration laden	75
Listing 7: TypoScript-Konfiguration laden	76

Inhalt der beiliegenden CD

Folgende Verzeichnisstruktur befindet sich auf beiliegender CD:

Verzeichnis	Beschreibung
DA/	Beinhaltet die Diplomarbeit in digitaler Form als PDF-Dokument.
Abbildungen/	Beinhaltet alle Abbildungen, die in der Diplomarbeit vorkommen
Weblinks/	Beinhaltet ein PDF zu jedem Weblink aus dem Literaturverzeichnis
i_search/	Fertige Erweiterung samt aller Dateien (inkl. Handbuch) sowie gepackter „.t3x“-Datei zum schnellen Import in jede TYPO3-Installation
required_extensions/	Beinhaltet Erweiterungen, die für die Funktion der Erweiterung „Intelligent Search Engine“ erforderlich sind.
xajax/	Beinhaltet die Extension „XAJAX“, die für die Funktion der Erweiterung „Intelligent Search Engine“ benötigt wird.
search_extensions/	Beinhaltet alle 4 Sucherweiterungen, die in Kapitel 4.2 untersucht wurden.
indexed_search/	Beinhaltet alle Dateien der Extension „indexed_search“ samt „.t3x“-Datei zum schnellen Import in jede TYPO3-Installation
gst_contentsearch/	Beinhaltet alle Dateien der Extension „gst_contentsearch“ samt „.t3x“-Datei zum schnellen Import in jede TYPO3-Installation

an_searchit/	Beinhaltet alle Dateien der Extension „an_searchit“ samt „.t3x“-Datei zum schnellen Import in jede TYPO3-Installation
mh_ajaxsearch/	Beinhaltet alle Dateien der Extension „mh_ajaxsearch“ samt „.t3x“-Datei zum schnellen Import in jede TYPO3-Installation