



JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis



Evaluierung, Auswahl und Customizing von Content Management Systemen zum Einsatz im Studiengang „Netzwerke und Sicherheit“

MAGISTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

in der Studienrichtung

INFORMATIK

Eingereicht von:

Karl Thier, 0356286

Angefertigt am:

FIM – Institut für Informationsverarbeitung und Mikroprozessortechnik

Betreuung:

o.Univ.Prof. Jörg R. Mühlbacher

Mitbetreuung:

DI Mag. Dr. Andreas Putzinger

Landfriedstetten, Oktober 2008

Kurzfassung

Diese Arbeit beschäftigt sich mit der Konzeptionierung und Umsetzung eines Internetportals für den Informatik-Masterstudiengang „Netzwerke und Sicherheit“ der Johannes Kepler Universität Linz.

Im ersten Teil der Diplomschrift werden die Grundlagen der Implementierung des Portals dargestellt. Nach der Analyse der Anforderungen an das Webportal erfolgt in diesem Teil auch ein Vergleich dreier, für die Realisierung in Frage kommender, Web Content Management Systeme.

Der zweite Teil ist auf das gewählte Web Content Management System fokussiert. Nach einer Vorstellung des gewählten Systems wird die Implementierung der Webseite beleuchtet.

Der dritte Teil der Arbeit gibt einen Überblick über die im Rahmen meines 10-stündigen Praktikums erarbeiteten Methoden, auf legale Weise Informationen über Internetnutzer zu sammeln.

Abstract

This diploma thesis is about the conceptual design and creation of an Internet Portal for the master study “Networks and Security” at Johannes Kepler University Linz.

In the first part of the thesis the basic principles of the portals’ implementation are described. After the analysis of the requirements for the portal three web content management systems, which possibly could be used for the implementation, are evaluated.

The second part is focused on the chosen web content management system. It deals with the features of the system and the concrete implementation of the portal.

The third part gives an overview of the methods to collect information about internet users legally, which had been discovered within my 10h practical.

Danksagung

Die Beendigung eines Studiums stellt den Abschluss eines alten und gleichzeitig natürlich auch den Anfang eines neuen Lebensabschnittes dar.

Solch ein Schnitt ist immer wieder eine gute Gelegenheit, zurück zu schauen und Danke zu sagen. Danke zu sagen, den vielen Menschen, die während des Studiums für mich da waren, sich mit mir gefreut haben und mich in schwierigeren Zeiten wieder aufgerichtet haben.

An erster Stelle sind hier natürlich meine Eltern zu nennen, die mir nicht nur das Studium an der Johannes Kepler Uni ermöglicht haben, sondern die auch und gerade in turbulenten Zeiten absolut hinter mir gestanden sind.

Auch meiner Schwester und meinen Freunden gilt besonderer Dank für ihre Unterstützung.

Die Idee des Security-Portals, die die Grundlage meiner Diplomarbeit darstellt, wurde am Institut für Informationsverarbeitung und Mikroprozessortechnik geboren, wo ich dieses Projekt schließlich auch in Angriff nehmen durfte.

Ich bedanke mich herzlich bei o. Univ. Prof Jörg R. Mühlbacher für die Betreuung dieser Diplomarbeit.

Den beiden Assistenten Dr. Michael Sonntag und DI Rudolf Hörmanseder gebührt Dank für ihre fachlichen und inhaltlichen Hilfestellungen während des Projektes.

Besonders bedanken möchte ich mich bei meinem Betreuer, DI Mag. Dr. Andreas Putzinger, der nicht nur in fachlicher Hinsicht, sondern auch sehr persönlich und mit viel Geduld ausgestattet, ein sehr wichtiger Wegbegleiter bei der Erstellung meiner Diplomarbeit war.

Euch allen gilt mein aufrichtiger Dank!

Inhaltsverzeichnis

Kurzfassung	ii
Abstract.....	iii
Danksagung.....	iv
Inhaltsverzeichnis	v
Abbildungsverzeichnis	x
1. Einleitung	1
1.1. Problemstellung und Motivation.....	1
1.2. Ziel der Arbeit	2
1.3. Gliederung der Arbeit.....	2
2. Anforderungsanalyse	3
3. Analyse und Vergleich von Content Management Systemen.....	4
3.1. (Web) Content Management System – Definition.....	4
3.2. Kriterienkatalog zum Vergleich von Content Management Systemen	5
3.2.1. Analyse aus Entwickler- bzw. Redakteurssicht	5
3.2.2. Sicherheit	6
3.2.3. Funktionen	7
3.2.4. Mehrsprachigkeit.....	8
3.3. Analyse und Vergleich ausgewählter Content Management Systeme	8
3.3.1. Einführung	8
3.3.2. Analyse aus Entwickler- bzw. Redakteurssicht	10
3.3.2.1. Installation des CMS	10
3.3.2.2. Komplexität der Bedienung	12
3.3.2.3. Erstellung von Webseiten – Konzept	15
3.3.2.4. Erweiterungskonzept.....	24
3.3.2.5. Features des Redaktionssystems	25
3.3.3. Sicherheit	28
3.3.3.1. Rechtesystem/Benutzermodell.....	28
3.3.3.2. Umgang mit Sicherheitslücken	31
3.3.3.3. CMS-interne Sicherheitskonzepte.....	33
3.3.3.4. Verbreitung des Systems, Referenzen.....	36
3.3.4. Funktionen	37

3.3.5.	Mehrsprachigkeit.....	38
3.4.	Auswahl eines CMS	40
3.4.1.	Typo3 – Vor- und Nachteile	40
3.4.2.	Drupal – Vor- und Nachteile	41
3.4.3.	Joomla! – Vor- und Nachteile.....	41
3.4.4.	Die Entscheidung	41
4.	Typo3 – Konzepte.....	42
4.1.	Die Ordnerstruktur einer Typo3-Installation	42
4.2.	Administration von Typo3	43
4.3.	TypoScript – die Typo3-eigene Skriptsprache	46
4.3.1.	Syntax	46
4.3.2.	Operatoren	47
4.3.3.	Sprachelemente in TypoScript	49
4.3.4.	Einsatz von TypoScript	50
4.3.4.1.	Templates.....	50
4.3.4.2.	PAGE-TSconfig	51
4.3.4.3.	USER-TSconfig	52
4.3.5.	Menüerstellung in TypoScript	52
4.4.	Seitenerstellung	53
4.5.	Templating mit TemplaVoila.....	56
4.5.1.	Struktur eines Templates	56
4.5.2.	Arten von Templates	57
4.5.3.	Die Erstellung eines Templates.....	57
4.5.4.	Einbindung von Templates in eine Seite	61
4.5.5.	Aktualisierung eines Templates	62
4.6.	Lokalisierung	62
4.6.1.	Übersetzung und Konfiguration einer Seite	62
4.6.2.	Übersetzung der Standard-Elemente	63
4.6.3.	Spracheinstellungen von TemplaVoila.....	64
4.6.4.	Übersetzung von Extensions.....	66
4.7.	Extensions – die Erweiterungen in Typo3	66
4.7.1.	Grundsätzliche Eigenschaften von Extensions	66
4.7.2.	Arten von Extensions	67
4.7.2.1.	Einteilung nach Installationsort	67

4.7.2.2.	Einteilung nach Funktionalität	67
4.7.3.	Die Verwaltung von Extensions	68
4.7.3.1.	Import extensions	69
4.7.3.2.	Install extensions/Loaded Extensions	69
4.7.3.3.	Translation handling.....	69
4.7.3.4.	Check for extension updates	69
4.7.3.5.	Settings	69
4.7.4.	Struktur einer Extension	70
4.7.5.	Erstellung einer Extension mit Hilfe von „Kickstarter“	71
4.7.6.	Erstellung der Programmlogik	73
4.7.7.	Mehrsprachigkeit in Extensions	74
4.7.8.	Die Verwendung von Ajax in Extensions	75
4.7.8.1.	Ajax-Request mit Hilfe von JavaScript.....	75
4.7.8.2.	Request in Typo3	76
5.	<i>security.fim.uni-linz.ac.at</i> – Konzept	76
5.1.	Gliederung von <i>security.fim.uni-linz.ac.at</i>	76
5.1.1.	Home	77
5.1.2.	Studium.....	77
5.1.3.	Sicherheit	78
5.1.4.	Forensik	78
5.2.	Design von <i>security.fim.uni-linz.ac.at</i>	78
5.2.1.	Header.....	78
5.2.2.	Auswahlbereich.....	78
5.2.3.	Hauptinhaltsbereich.....	78
5.2.4.	Fußzeile	79
6.	Implementierung.....	80
6.1.	Seitenstruktur	80
6.2.	Konfiguration – das Root-Template.....	81
6.3.	Das Seitentemplate.....	82
6.3.1.	Die HTML-Vorlage.....	83
6.3.2.	Erstellung des Seitentemplates	84
6.3.3.	Koppelung des Seitentemplates an den Seitenbaum	85
6.4.	Inhalt.....	86
6.4.1.	Die Portalseiten	86

6.4.2.	Statische Informationen.....	87
6.4.3.	Das Team.....	87
6.4.4.	Lehrveranstaltungen.....	89
6.5.	Erstellung von Menüs.....	90
6.5.1.	Das Hauptmenü.....	90
6.5.2.	Rootline.....	92
6.5.3.	Schnellwahlmenü.....	92
6.5.4.	Seitliches Menü.....	94
6.6.	Extensions.....	94
6.6.1.	tt_news.....	94
6.6.1.1.	Konfiguration.....	95
6.6.1.2.	Newskategorien.....	96
6.6.1.3.	Newseinträge.....	97
6.6.1.4.	News-Plugin.....	98
6.6.2.	xml_ttnews_import.....	102
6.6.2.1.	Installation und Konfiguration.....	102
6.6.2.2.	Änderung der Extension mit Hilfe des Kickstarters.....	103
6.6.3.	„realurl“.....	105
6.6.3.1.	Konfiguration.....	105
6.6.4.	indexed_search + crawler.....	106
6.6.4.1.	indexed_search.....	106
6.6.4.2.	crawler.....	107
6.6.5.	kj_imagelightbox2.....	109
6.6.6.	mailformplus, kt_student_file_system.....	109
6.6.6.1.	Mailformplus.....	110
6.6.6.2.	kt_student_file_system.....	113
6.6.7.	kt_kusss_date_export.....	118
6.6.7.1.	Voraussetzungen.....	119
6.6.7.2.	Die Implementierung.....	119
6.6.8.	kt_security_widgets.....	120
6.6.8.1.	Implementierung.....	120
6.6.8.2.	Die Widgets.....	122
7.	Integration des 10h Praktikums „Gläserner Benutzer“ in <i>security.fim.uni-linz.ac.at</i>	122
7.1.	Aufgabenstellung.....	122

7.2. Möglichkeiten der Informationssammlung.....	123
7.2.1. TCP/IP-Verbindung.....	123
7.2.2. HTTP-Request	123
7.2.3. Java-Applets.....	124
7.2.4. JavaScript.....	124
7.2.4.1. Fingerprinting mit JavaScript	125
7.2.4.2. JavaScript + CSS.....	125
7.3. Integration in <i>security.fim.uni-linz.ac.at</i>	126
8. Fazit und Ausblick	126
9. Literatur.....	128
Eidesstattliche Erklärung	132

Abbildungsverzeichnis

Abbildung 1: Backendansicht eines Typo3-Systems	13
Abbildung 2: Backend eines frisch installierten Joomla!-Systems	14
Abbildung 3: Die Oberfläche von Drupal.....	15
Abbildung 4: Seitenbaum in Typo3	53
Abbildung 5:Konfiguration einer neuen Seite	54
Abbildung 6: Erstellung der Data Structure in TemplaVoila	57
Abbildung 7: Mapping eines Feldes in TemplaVoila.....	60
Abbildung 8: Übersetzung von Content Elements	63
Abbildung 9: Sprachvererbung in TemplaVoila	65
Abbildung 10: Verschiedene Sprachversionen eines Plugins	65
Abbildung 11: Extension Manager im Typo3 Backend	68
Abbildung 12: Seitenstruktur von <i>security.fim.uni-linz.ac.at</i>	79
Abbildung 13: Seitenbaum in Typo3	80
Abbildung 14: Inhalte der Startseite.....	86
Abbildung 15: RichText Editor mit Teilen des Inhaltes von "Voraussetzungen"	87
Abbildung 16: Personen-Vorlage.....	89
Abbildung 17: Das Hauptmenü.....	90
Abbildung 18: Rootline	92
Abbildung 19: Schnellauswahlmenü	92
Abbildung 20: Menü am linken Seitenrand	94
Abbildung 21: Erstellung von News feed imports	102
Abbildung 22: Bearbeitung einer Extension im Kickstarter.....	103
Abbildung 23: Oberfläche von Kickstarter.....	114
Abbildung 24: Die von Kickstarter erzeugten Dateien	116
Abbildung 25: Security-Widgets.....	120
Abbildung 26: HTTP-Request eines Webbrowsers	123
Abbildung 27: URL einer Google-Suche	126

1. Einleitung

Diese Diplomarbeit beschäftigt sich mit der Konzeption und der anschließenden Implementierung eines Webportals für den Masterstudiengang „Netzwerke und Sicherheit“ an der Johannes Kepler Universität Linz. Im Entwurfsprozess des Webportals ist eine Evaluierung verschiedener Web Content Management Systeme nach einem zuvor zu definierenden Kriterienkatalog vorgesehen. Die Implementierung erfolgt auf Basis des am besten geeigneten Web Content Management Systems. In diesem einleitenden Kapitel wird ein erster Überblick über die Aufgabenstellung gegeben und die Gliederung dieser Arbeit erläutert.

1.1. Problemstellung und Motivation

Im Zuge der Studienplanreform des Jahres 2007 wurden an der Johannes Kepler Universität Linz neben den bisherigen Informatik-Masterstudien „Informatik“ sowie der fächerübergreifenden „Bioinformatik“ drei weitere Masterstudiengänge installiert. Es sind dies „Pervasive Computing“, „Software Engineering“ und „Netzwerke und Sicherheit“.

Am Institut für Informationsverarbeitung und Mikroprozessortechnik (FIM), welches die Mehrheit der Pflichtlehrveranstaltungen für „Netzwerke und Sicherheit“ anbietet, wurde daraufhin die Idee einer Informationsplattform geboren. Diese sollte neben studienspezifischen Informationen, wie Details zum Studienplan, zu den Lehrveranstaltungen und Lehrenden, etc. auch eine Anlaufstelle für alle fachlich Interessierten werden.

Die Bereitstellung aktueller Entwicklungen („News“) in sicherheitsrelevanten Themen, ergänzt um Hintergrundberichte und Artikel, gewährleisten dies.

Besonders wichtig an der Plattform ist, dass sie auch von Studierenden mit Inhalten versehen werden kann. Jeder Student soll die Möglichkeit haben, seine Arbeiten auf dem Portal zu veröffentlichen und damit Kolleginnen und Kollegen und anderen Interessierten zur Verfügung zu stellen. So ist sichergestellt, dass in Seminararbeiten und Praktika angehäuften Wissen nicht verloren geht, sondern an einer zentralen Stelle für alle verfügbar ist.

1.2. Ziel der Arbeit

Diese Arbeit verfolgt zwei Ziele. Einerseits soll durch den Vergleich von mehreren Web Content Management Systemen an Hand eines zu definierenden Kriterienkataloges die Auswahl des für die Plattform verwendeten Web Content Management Systems argumentiert werden. Zugleich ist diese Evaluierung auch eine Abhandlung über mögliche Vergleichskriterien für Web Content Management Systeme.

Andererseits wird die praktische Erstellung der Informationsplattform dokumentiert. So soll dem Leser die theoretische Möglichkeit gegeben werden, die konkrete Implementierung selbstständig zu warten und selbst weiter zu entwickeln.

1.3. Gliederung der Arbeit

Im nachfolgenden Kapitel werden in der Anforderungsanalyse die an das Informationsportal gestellten Anforderungen erhoben und definiert.

Der dritte Hauptabschnitt ist dem Vergleich der Web Content Management Systeme gewidmet. Im ersten Teil werden die Kriterien, welche zum Vergleich herangezogen werden, an Hand der Anforderungsanalyse definiert. Anschließend erfolgt der Vergleich dreier Web Content Management Systeme und abschließend die Auswahl des am besten geeigneten.

Im darauf folgenden Kapitel wird das ausgewählte System aus konzeptueller Sicht genauer vorgestellt.

Die Konzeption und das Design der Informationsplattform als solche sind Bestandteile des fünften Kapitels dieser Arbeit.

Anschließend wird die Implementierung, also die Realisierung des Konzeptes mit Hilfe des ausgewählten Web Content Management Systems näher erläutert.

Im siebten Kapitel wird die Integration des 10h-Praktikums des Verfassers in die Webseite dargestellt. Dieses untersucht unter dem Titel „Gläserner Nutzer“ Möglichkeiten der impliziten Informationsgewinnung über Benutzer auf Webseiten.

Abschließend erfolgt eine Evaluierung und Bewertung der eigenen Arbeit durch den Autor.

2. Anforderungsanalyse

Die Anforderungen an das zu erstellende Informationsportal werden im folgenden Kapitel im Rahmen der Anforderungsanalyse erhoben. Ausgehend vom Ziel „ein Webportal für das Masterstudium ‚Netzwerke und Sicherheit‘ zu erstellen“ wurden die notwendigen Funktionen und Eigenschaften in Gesprächen mit den zukünftigen Betreibern erhoben und durch eigene Vorschläge und Ideen in Absprache mit diesen ergänzt.

Die Informationsplattform soll der breiten Öffentlichkeit im Internet zugänglich sein und unter anderem einen „Werbeeffekt“ für das Studium „Netzwerke und Sicherheit“ bieten. Da bereits das Studium mit den Schlagworten „internet security“ bzw. „network security“ beworben wird, soll auch das Portal mit der URL „*security.fim.uni-linz.ac.at*“ bekannt gemacht werden.

„*security.fim.uni-linz.ac.at*“ soll die erste Anlaufstelle für Studierende sein, die sich einen Überblick über das Studium verschaffen möchten. Die Informationen sollen gleichzeitig kompakt, interessant und doch umfassend sein und die Möglichkeiten, die das Studium eröffnet, wie auch das Qualifikations- und Anforderungsprofil skizzieren.

Dieses Angebot wird durch tagesaktuelle Informationen zu den Lehrveranstaltungen und zum Studium abgerundet, wobei das Portal in keiner Weise als Konkurrenz zum Studieninformationssystem „KUSSS“ gedacht ist. Die Schnittstellen, die KUSSS zum Datenexport anbietet, werden genutzt, um redundante Datenhaltung zu vermeiden. So werden zum Beispiel die Kursdaten aus dem KUSSS exportiert und auf dem Portal angezeigt.

Ein weiteres wichtiges Standbein des Portals sind die fachspezifischen Inhalte. Die Plattform bietet tagesaktuelle Neuigkeiten zu jenen Themen, deren theoretische und praktische Grundlagen im Studium erarbeitet und vertieft werden. Diese Informationen werden durch Grundlagenartikel, wissenschaftliche Publikationen und Arbeiten von Studenten ergänzt.

Das Studium „Netzwerke und Sicherheit“ deckt ein breites Spektrum an Themen ab. Diese reichen von Computerforensik und rechtlichen Fragestellungen der IT über aktuelle Algorithmen zur Verschlüsselung bis hin zur Einrichtung von sicheren Netzwerken in Betrieben. Um dieser Vielzahl an Themen Rechnung zu tragen, ist das Portal thematisch in „Unterportale“ gegliedert, welche ihrerseits eigene Startseiten sowie fachspezifische Inhalte anbieten. Jede im Studium angebotene Pflichtlehrveranstaltung ist thematisch einem der Unterportale zugeordnet. Der Bereich „Studium“ wird als eigenes thematisches Unterportal geführt.

Jedes Jahr werden von Studenten Arbeiten in Seminaren und Praktika angefertigt. Diese Arbeiten sollen Interessierten zugänglich gemacht werden. Dazu bietet das Portal ein System, welches den Upload von Arbeiten, gebunden an Lehrveranstaltungen, ermöglicht. Um Missbrauch zu vermeiden, wird ein Freigabesystem eingerichtet. Dieses gibt dem Lehrveranstaltungsleiter die Möglichkeit, Arbeiten vor der Veröffentlichung zu prüfen und per Mausklick entweder abzulehnen oder freizuschalten.

Um die Aktualität der Seite zu gewährleisten, ohne die redaktionelle Arbeitsbelastung für die Betreuer des Portals zu groß werden zu lassen, wird ein System zur automatischen Sammlung und Aktualisierung von Nachrichten aus frei zugänglichen Quellen benötigt.

Das Internetportal soll der internationalen Ausrichtung der Johannes Kepler Universität gerecht werden. Das Angebot des Portals in Deutsch und Englisch ermöglicht es, *security.fim.uni-linz.ac.at* auch ausländischen Studierenden zur Verfügung stellen zu können.

Um nicht bereits bestehende Funktionalität aufwändig neu entwickeln zu müssen, soll das Portal in ein geeignetes Web Content Management System eingebettet werden.

Die Auswahl des Web Content Management Systems ist explizit Teil des Diplomprojektes. Der Vergleich verschiedener in Frage kommender Content Management Systeme und die schlussendliche Auswahl des am besten geeigneten werden in Kapitel 3 dieser Arbeit eingehend behandelt.

3. Analyse und Vergleich von Content Management Systemen

Die Auswahl des am besten geeigneten Web Content Management Systems soll nach eingehender Analyse verschiedener in Frage kommender Systeme an Hand eines vorher festgelegten Kriterienkatalogs erfolgen.

Der erste Teil dieses Kapitels ist der Erstellung dieses Kriterienkatalogs gewidmet. Anschließend werden drei in Frage kommende Web Content Management Systeme an Hand des Katalogs miteinander verglichen. Das Kapitel schließt mit der konkreten Auswahl des am besten geeigneten Web Content Management Systems.

3.1. (Web) Content Management System – Definition

Web Content Management Systeme (kurz WCMS) ermöglichen die gemeinschaftliche Erstellung und Bearbeitung der Inhalte eines webbasierten Informationssystems, ohne, dass die Benutzer Kenntnisse von den dahinterliegenden Technologien, wie z.B. PHP, HTML oder ASP.NET, haben müssen. Der hier verwendete Begriff „Web Content Management System“ wird im allgemeinen Sprachgebrauch durch den fachlich etwas weiter definierten Begriff

„Content Management System“ (CMS) ersetzt. Der Unterschied zwischen einem WCMS und einem CMS besteht in der Notwendigkeit eines webbasierten Informationssystems. Während ein Content Management System nicht zwingend webbasiert sein muss, ist dies bei einem WCMS eine Voraussetzung. Auf Grund der allgemein üblichen Verwendung des Begriffes CMS für WCMS wird auch in der vorliegenden Arbeit von Content Management Systemen gesprochen. [WIK-07]

CMS erleichtern die Redaktionsarbeit für Webseiten mit sich schnell ändernden Inhalten und sorgen gleichzeitig für mehr Sicherheit, da die Systeme ein Benutzer- und Rechtemodell anbieten. Ein Redakteur kann durch ein CMS so z.B. nur Inhalte erstellen oder ändern und nicht löschen, während er ohne CMS bzw. ohne Rechteverwaltung möglicherweise sogar unabsichtlich die ganze Webseite löschen könnte.

Die meisten heutigen Content Management Systeme bieten die Möglichkeit, Inhalt getrennt von Struktur und Layout zu warten. Dadurch ergibt sich die Möglichkeit der Änderung von Layout und Struktur einer Webseite, ohne dass die Inhalte angepasst werden müssten. Eine Neugestaltung von inhaltlich umfangreichen Seiten wird dadurch wesentlich erleichtert. [TYP-07/1].

3.2. Kriterienkatalog zum Vergleich von Content Management Systemen

Der zum Vergleich der CMS erarbeitete Kriterienkatalog beleuchtet verschiedene wichtige Merkmale von Content Management Systemen wie Sicherheit, Benutzerfreundlichkeit oder Erweiterungsmöglichkeiten.

Der Kriterienkatalog ist allgemein und daher nicht nur für dieses Projekt anwendbar. Der Einfluss der einzelnen Kriterien auf die Entscheidungsfindung ist bei jedem Projekt trotzdem neu zu bewerten.

3.2.1. Analyse aus Entwickler- bzw. Redakteurssicht

- *Installation des CMS*

Mit einem kurzen Überblick über die Systemvoraussetzungen sowie über die Vorgehensweise bei der Installation des jeweiligen Content Management Systems startet der Vergleich.

- *Komplexität der Bedienung*

Dieses Kriterium vergleicht die Lernkurve des Redaktionssystems, die Intuitivität der Benutzung und die Komplexität des Gesamtsystems. Hier wird z.B. berücksichtigt, ob Features, die ein Redakteur nicht benötigt, mit Hilfe des Rollenmodells ausgeblendet werden können, um die Bedienung so einfach wie möglich zu gestalten.

- *Erstellung von Struktur, Inhalt und Layout einer Webseite*

Hier sollen Konzepte und Möglichkeiten der Systeme zur Erstellung von Webseiten verglichen werden.

- *Erweiterungskonzepte*

Die meisten Content Management Systeme bieten die Möglichkeit, nicht vorhandene Funktionalität in Form von Erweiterungen hinzuzufügen. Welche Möglichkeiten dies konkret sind, wird in diesem Punkt vorgestellt.

- *Features des Redaktionssystems*

Hier sollen alle Features präsentiert werden, welche dem Editor und Redakteur die Arbeit erleichtern. Dazu zählen zum Beispiel Editoren, welche die direkte Bearbeitung und Formatierung von Texten ermöglichen (so genannte WYSIWYG-Editoren, von „What you see is what you get“), weiters die zeitgesteuerte Veröffentlichung von Artikeln oder auch die automatische Archivierung nach einer gewissen Zeitspanne.

3.2.2. Sicherheit

Gerade bei einer Webseite zum Thema „IT-Sicherheit“ muss angenommen werden, dass diese von Hackern und Crackern verstärkt auf Sicherheitsmängel und -lücken geprüft wird und ein begehrtes Ziel von Attacken und Angriffen ist.

Das gewählte CMS sollte daher „so sicher als möglich“ sein, sowohl vom Gesichtspunkt des grundsätzlichen Designs als auch vom Gesichtspunkt der tatsächlichen Implementierung des Systems. Im Wissen, dass absolute Sicherheit praktisch nicht realisierbar ist, werden folgende Kriterien zum Vergleich herangezogen:

- *Rechtekonzept und Benutzermodell*

„*security.fim.uni-linz.ac.at*“ wird von verschiedenen Personen administriert. Jeder Lehrveranstaltungsleiter soll die Informationen zu seinen Lehrveranstaltungen bearbeiten können, die verschiedenen Portalseiten werden von Wissenschaftlern aus dem jeweiligen Fach betreut. Außerdem sollen News gepostet werden können und anderes mehr.

Bewertet werden hier erstens das grundsätzliche Vorhandensein eines Benutzer- und Rechtesystems sowie die Anpassungsmöglichkeiten dieses Systems. Ein eventuell vorhandenes Rollenmodell wird analysiert und die Flexibilität und Granularität des Systems durchleuchtet.

- *Anzahl der installierten Systeme, bekannte Systeme*

Die Anzahl und die Art der Benutzer ist ein weiterer Indikator für die Sicherheit eines CMS. Ein oft eingesetztes System wird automatisch häufiger angegriffen und damit

auf Sicherheit getestet als ein unbekanntes Produkt. Weiters ist die Community bei einem häufig verwendeten CMS automatisch größer, was die Chancen, dass Sicherheitslücken entdeckt werden, erhöht.

Wird ein CMS von vielen großen Unternehmen und Institutionen genutzt, so ist dies ein Hinweis auf besonderes Vertrauen in das System.

- *Umgang mit Sicherheitslücken*

Besonders wichtig im Hinblick auf die Sicherheit eines Content Management Systems ist die Reaktion auf bekannt gewordene Mängel und Angriffsmöglichkeiten. Die Regelmäßigkeit von Updates wird ebenso beleuchtet wie die Zeitdauer, bis eine Sicherheitslücke behoben ist. Schlussendlich wird analysiert, ob die Reaktion auf Sicherheitslücken und Fehler in einen definierten Prozess eingebettet ist.

- *Zusätzliche Sicherheitsmechanismen*

Hier sollen CMS-eigene Sicherheitsfeatures vorgestellt und verglichen werden.

3.2.3. Funktionen

Wie in Abschnitt 2 erörtert, wird *security.fim.uni-linz.ac.at* viele verschiedene Funktionen bieten. Idealerweise unterstützt das zu verwendende CMS alle diese Funktionen.

Anwendungen, welche allerdings nicht standardmäßig im System integriert sind, müssen hinzugefügt bzw. selbst erstellt werden. Dieser Abschnitt des Kriterienkataloges beschäftigt sich daher mit der Basisfunktionalität der zu vergleichenden Systeme, sowie notwendigen Erweiterungen.

Das begutachtete Content Management System wird speziell auf das Vorhandensein folgender Features geprüft:

- Newssystem
- Übernahme von RSS- bzw. Atom-Feeds externer Webseiten in das eigene News-System
- Volltextsuche
- Automatische Bildverarbeitung (Stichwort „Thumbnails“)
- Möglichkeit zum Dateiupload
- Workflowsystem zur Unterstützung des Dateiuploads

3.2.4. Mehrsprachigkeit

„security.fim.uni-linz.ac.at“ wird in den Sprachen Deutsch und Englisch zur Verfügung stehen. Die von den Content Management Systemen angebotenen Konzepte zur Mehrsprachigkeit werden erläutert, verglichen und auf Tauglichkeit für die Zwecke des Portals überprüft.

3.3. Analyse und Vergleich ausgewählter Content Management Systeme

Der folgende Abschnitt dieses Kapitels ist dem Vergleich von drei Content Management Systemen nach den oben angeführten Kriterien gewidmet.

CMS sind in sehr großer Zahl verfügbar. Auf cmsmatrix.org werden derzeit 933 verschiedene (Stand 27. Juli 2008), zum Teil sehr unterschiedliche Content Management Systeme gelistet. [CMS-07].

Die schiere Anzahl an Content Management Systemen zwingt zu einer Vorauswahl. Es wurden nach einer Prüfung, drei allgemein bekannte, geeignet scheinende Content Management Systeme ausgewählt. Es sind dies Drupal, Typo3 sowie Joomla!.

Nach einer kurzen Vorstellung jedes Systems werden die Punkte des Kriterienkatalogs in Abschnitt 3.2 der Reihe nach abgearbeitet. Jedes Hauptkriterium wird mit einer kurzen Zusammenfassung abgeschlossen.

3.3.1. Einführung

Im Folgenden werden die ausgewählten CMS kurz vorgestellt.

Typo3

Typo3 ist ein PHP-basiertes Content Management System. Es kann als eines der ersten CMS überhaupt bezeichnet werden, da die Entwicklung begann, als der Begriff „Content Management System“ noch weithin unbekannt bzw. überhaupt noch nicht erfunden war, nämlich im Jahre 1997. Untrennbar ist ein Name mit Typo3 verbunden: der Name Kasper Skårhøj.

Ein Kunde des Dänen benötigte ein System, welches die getrennte Bearbeitung des Inhalts seiner Webseite von Layout und Struktur ermöglichen sollte. Bereits wenig später, zu Beginn 1998, wurden drei Prototypen vorgestellt. Im Sommer des gleichen Jahres wurde die Entwicklung von superfish.com kommerzialisiert. Im Oktober 1998 wurde Typo3 im Rahmen eines Workshops einer breiteren Öffentlichkeit vorgestellt. Basierend auf Kundenrückmeldungen wurde im Jänner 1999 die Entwicklung eines komplett neuen Anwendungskerns beschlossen.

Die Entscheidung, Typo3 zu kommerzialisieren, wurde bereits im Sommer 1999 wieder revidiert – Kasper Skårhøj sicherte sich alle Rechte für Typo3, verließ superfish.com und trieb die Entwicklung ein Jahr alleine voran. Die erste öffentliche Beta-Version von Typo3 wurde im August 2000 veröffentlicht.

Rasch wuchs die Community um das Content Management System und im April 2001 fand der erste Typo3-Workshop statt. Obwohl immer noch in der Beta-Phase befindlich, wurde Typo3 von der Fachpresse bereits als ernst zu nehmendes und ausgereiftes Produkt gesehen.

Die erste „finale“ Version von Typo3, Version 3.0, wurde am 24. Mai 2002 veröffentlicht.

Der Extension-Manager, welcher für die Anpassung und Erweiterung der Typo3-Funktionalität besonders wichtig ist, wurde im November 2002 veröffentlicht. Mittlerweile ist die nächste Version, nämlich 4.0 veröffentlicht. Im April dieses Jahres wurde Typo3 4.2 präsentiert.

Die Arbeiten an der Version 5.0 haben längst begonnen. [TYP-07/2]

Typo3 4.x.x wird von verschiedenen Teams weiter entwickelt, wobei ein „Core-Team“ von momentan 26 Personen hauptverantwortlich ist. Unterstützt wird das Core-Team dabei vom „Content Rendering Team“, zuständig für die Implementierung der XHTML-konformen Ausgabe des gerenderten Inhalts. Daneben gibt es noch das „Extension Coordination Team“, dessen Mitglieder aber primär unterstützende Funktion für Entwickler von Erweiterungen haben und das „Design-Team“, zuständig für die Corporate Identity von Typo3. Für die Entwicklung der Version 5.0 gibt es ebenso ein eigenes Team, ein anderes Team ist für die Sicherheit des Codes zuständig.[TYP-07/3]

Typo3 wird unter der „General Public License“ angeboten. Die unten angeführten Tests und Kommentare beziehen sich auf die Version 4.2.1.

Joomla!

Joomla! ist ein PHP-basiertes CMS. Das Content Management System ist eine Abspaltung des bekannten CMS „Mambo“. Die Besitzerin von Mambo, die Firma „Miro International Pty Ltd“, entschloss sich aus finanziellen und rechtlichen Gründen im August 2005, Mambo als Marke schützen zu lassen. [TVK-08]

Die meisten Entwickler von Mambo, die in diesem Schritt von Miro den open-source Gedanken verletzt sahen, entschlossen sich, auf Basis von Mambo ein neues CMS zu erstellen. Bereits im September 2005 wurde von Joomla! die Version 1.0.0 freigegeben. Im Dezember 2007 wurde die neue, auf PHP5 basierende Version 1.5 von Joomla! vorgestellt. Beide Versionen werden bis heute von einem Kernteam weiterentwickelt, unterstützt durch

Arbeitsgruppen, welche Kommunikationsschnittstelle zwischen dem Kernteam und der Community sind. Spezielles Anliegen des Kernteams ist, bei maximaler Funktionalität die Bedienung von Joomla! so einfach als möglich zu halten. [JOO-08/1]

Joomla! wird unter der „GNU-General Public License“ angeboten.

Die unten angeführten Angaben und Tests beziehen sich auf Joomla! 1.5.2.

Drupal

Drupal ist ein Content Management System, welches auf PHP basiert. Obwohl Drupal eigentlich für alle Arten von Webseiten eingesetzt werden kann, liegt der Schwerpunkt auf Web 2.0-Anwendungen.

Diese Ausrichtung von Drupal ist aus der Geschichte heraus zu verstehen, da der Vorläufer von Drupal von einigen niederländischen Studenten als einfaches Newsboard zum Austausch von Nachrichten programmiert wurde. Später wurde das Board weiterentwickelt und im Internet unter der Domain drop.org zu einem Ort der Diskussion über neue Web-Technologien, welche auf der Plattform dann auch gleich implementiert und getestet wurden. Im Jänner 2001 veröffentlichten die Betreiber das hinter drop.org liegende System als Content Management System unter dem Namen Drupal. Diese Bezeichnung ist von der englischen Aussprache des niederländischen Wortes für Tropfen, „druppel“, abgeleitet.

Drupal wird unter der „General Public License“ angeboten. Die weiteren Tests beziehen sich auf die Drupal-Version 6.2.

3.3.2. Analyse aus Entwickler- bzw. Redakteurssicht

3.3.2.1. Installation des CMS

Typo3

Typo3 bietet zwei Möglichkeiten der Installation.

Hat man bereits einen Webserver inklusive Datenbank aufgesetzt, so installiert man Typo3 durch Kopieren des Quellcode-Packages und des sogenannten „Dummy-Packages“, welches das Grundgerüst für eine neue Seite enthält, auf den Webserver. Beide Packages werden im Internet als zip-Archiv angeboten. Zur leichteren Einarbeitung kann man noch zusätzlich die „Quickstart“ und „Testsite“-Packages installieren, welche vorgefertigte Beispiel-Webseiten enthalten.

Als Webserver für eine typische Typo3 Installation wird meist Apache eingesetzt, es können jedoch auch andere Webserver mit PHP-Funktion eingesetzt werden.

In Typo3 werden sämtliche Inhalte sowie die Struktur der verwalteten Webseiten sowie alle Benutzer von Typo3, in einer Datenbank gespeichert. Standardmäßig ist als Datenbank „MySQL“ vorgesehen, für die Typo3-Versionen seit 4.0 gibt es allerdings eine Erweiterung (Extension), welche als Datenbankabstraktionsschicht fungiert. Dadurch wird auch die Verwendung von anderen Datenbanksystemen wie „Oracle“ und „PostgreSQL“ ermöglicht. [DAM-06].

Ruft man Typo3 zum ersten Mal auf, so muss man noch die Datenbankanbindung konfigurieren, was aber durch einen integrierten Assistenten sehr vereinfacht wird. Dieser Assistent, in Typo3 „Install Tool“ genannt, steht auch nach der Installation zur Verfügung und ermöglicht die Überprüfung der Funktionalität der Typo3-Installation.

Hat man keinen Webserver installiert bzw. möchte man Typo3 lokal testen, so gibt es auch komplette Installer-Packages, welche Apache und MySQL inklusive Typo3 installieren. Diese Art der Installation erleichtert den Einstieg in Typo3, da die Konfiguration des Servers und auch die grundlegenden Einstellungen von Typo3 automatisch erfolgen und sofort mit der eigentlichen Arbeit in Typo3 begonnen werden kann. Aus Sicherheitsgründen wird diese Art der Installation im Produktivbetrieb jedoch nicht empfohlen.

Joomla!

Joomla! wird als einfaches zip-Archiv zum Download angeboten. Die Installation erfolgt durch Entpacken dieses Archivs auf den Webserver. Offiziell wird der Apache-Webserver empfohlen. Es wird eine Datenbank zur Datenhaltung benötigt – derzeit wird nur MySQL offiziell unterstützt.

Ein webbasierter Assistent führt durch die grundlegenden Konfigurationsschritte von Joomla!, wie Datenbankanbindung, Erstellung eines Administratorkontos und Spracheinstellungen. Joomla 1.5 ist komplett auf Deutsch verfügbar. [JOO-08/2]

Auf Wunsch können auch Beispieldateien installiert werden, welche grundlegende Informationen zu Joomla! vermitteln sollen. Als Sicherheitsmaßnahme muss nach Abschluss der Installation der Installationsordner im Joomla-Verzeichnis komplett gelöscht werden. Ohne diese Maßnahme kann Joomla! nicht verwendet werden.

Drupal

Der Quellcode von Drupal wird als Archiv zum Download angeboten und auf den gewünschten Webserver entpackt. Für den Betrieb von Drupal wird die Verwendung von Apache als Webserver und MySQL als Datenbank empfohlen. Auf der Webseite wird jedoch

explizit darauf hingewiesen, dass der Betrieb auch mit dem „Internet Information Server“ von Microsoft möglich ist. Als Datenbank kann auch PostgreSQL eingesetzt werden. [DRU-08/1] Die Installation gestaltet sich durch den integrierten Assistenten einfach. Neben der Datenbankanbindung muss ein Administratorkonto eingerichtet werden, ein Seitenname vergeben sowie die Zeitzone des Servers angegeben werden. Als einziges der getesteten CMS bietet Drupal eine Information über die Stärke des gewählten Administratorpassworts und empfiehlt bei Bedarf, ein stärkeres zu verwenden. Drupal ermöglicht auch eine automatische Benachrichtigung per Mail, wenn neue Releases zum Download bereitstehen.

3.3.2.2. Komplexität der Bedienung

Typo3

Typo3 kann komplett webbasiert verwaltet werden. Neben der eigentlichen Webseite, dem sogenannten „Frontend“, kann auch die Administrationsoberfläche, das „Backend“, über das Web aufgerufen werden und zwar über die Adresse *<Server-Adresse>/typo3*.

Der besseren Bedienbarkeit wegen, wird die Trennung zwischen Frontend und Backend nicht ganz strikt eingehalten. Wenn im Frontend eine Login-Seite vorhanden ist, kann sich ein Backend-Benutzer auch im Frontend anmelden. Diese Anmeldung ermöglicht dem Benutzer die direkte Änderung von Inhalten im Frontend. Ein angemeldeter Backend-Benutzer ist auch im Frontend automatisch angemeldet. Grundlegende Konfigurationsschritte müssen trotzdem im Backend getätigt werden.

Das Backend ist dreigeteilt. Im linken Bereich findet sich das sehr umfassende Menü, rechts oben sind seit der Version 4.2 häufig benutzte Funktionen zu finden, wie das Löschen der Caches oder die Logout-Funktion. Rechts unten werden die Inhalte zum ausgewählten Menüpunkt angezeigt. Dieser Bereich wird je nach ausgewähltem Menüpunkt nochmals geteilt. Im linken Feld wird dann die Gliederung der Webseite in Baumform angezeigt, rechts die Inhalte der jeweiligen Seite korrespondierend zum Menüpunkt. Die Anordnung der Funktionen erscheint logisch und durchdacht. Vor allem die Teilung des rechten Feldes, mit der über den Seitenbaum der Inhalt jeder Seite schnell zugänglich gemacht wird, erleichtert die Arbeit ungemein.

Die graphische Benutzeroberfläche von Typo3 erscheint sehr „maschinell“ und technisch. Man erkennt deutlich, dass die Zielgruppe von Typo3 professionelle Programmierer sind. Die Bedienung von Typo3 aus Sicht eines Webentwicklers erfordert eine gewisse, nicht zu unterschätzende Einarbeitungszeit.

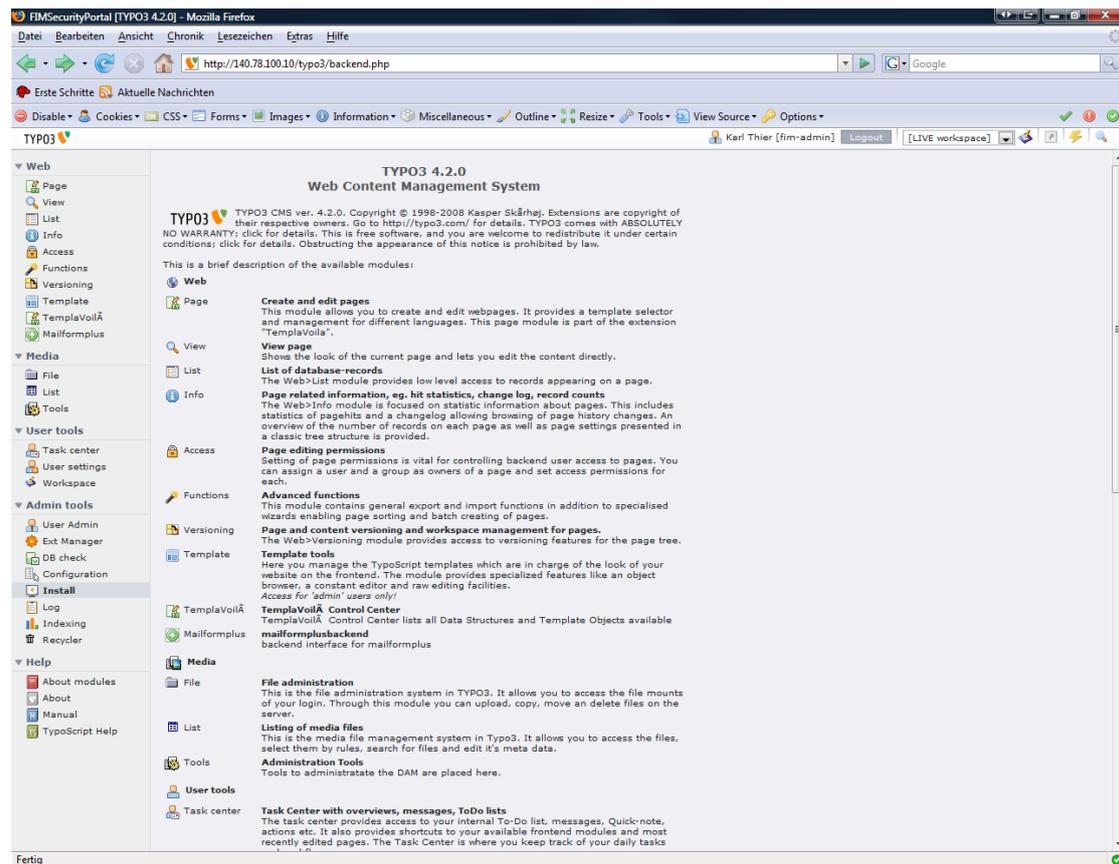


Abbildung 1: Backendansicht eines Typo3-Systems

Das Menü besitzt zwei Ebenen. Alle Menüeinträge sind immer sichtbar. Dies sorgt zwar für schnelle Auswählbarkeit, gleichzeitig irritiert die schiere Menge an Menüpunkten anfangs etwas. Für jeden Backend-Benutzer lässt sich daher das Backend so konfigurieren, dass nur die benötigten Funktionen sichtbar sind. Dies mindert die Komplexität doch deutlich. Außerdem befindet sich auf der Startseite des Backends eine Übersicht über alle Menüpunkte mit einer kurzen Beschreibung der Funktion.

Joomla!

Auch in Joomla! wird zwischen Frontend als eigentliche Webseite und Backend als Administrationsoberfläche unterschieden. Die Administrationsoberfläche kann unter `<server>/administrator` aufgerufen werden.

Aus Gründen der besseren Bedienbarkeit ist es auch in Joomla! möglich, Inhalte im Frontend zu bearbeiten. Das Backend erscheint frisch und farbenfroh und mit großen, bunten Icons. Viel wesentlicher ist allerdings, dass die Funktionen gut und durchdacht angeordnet sind. Im oberen Bereich des Backends befindet sich die Menüleiste, über die jederzeit auf alle Funktionen zugegriffen werden kann. Die Untermenüpunkte sind nur dann sichtbar, wenn sich der Mauszeiger über den Hauptmenüpunkt bewegt. Dies ist allerdings nicht störend, da

die Hauptmenüpunkte sprechend benannt sind und Rückschlüsse auf die enthaltenen Funktionen zulassen. Rechts neben der Menüleiste befinden sich die Vorschaufunktionen, der Logout-Button sowie die Anzahl der gerade angemeldeten Benutzer und die Anzahl der Nachrichten im Joomla!-eigenen Posteingang. Unterhalb der Menüleiste wird der Inhalt zum ausgewählten Menüpunkt angezeigt. In Joomla! ist es nicht möglich, die Administrationsoberfläche benutzerspezifisch zu konfigurieren. Alleine die Zugehörigkeit zu einer der drei Backendbenutzergruppen entscheidet, welche Menüpunkte man sieht und auswählen kann.

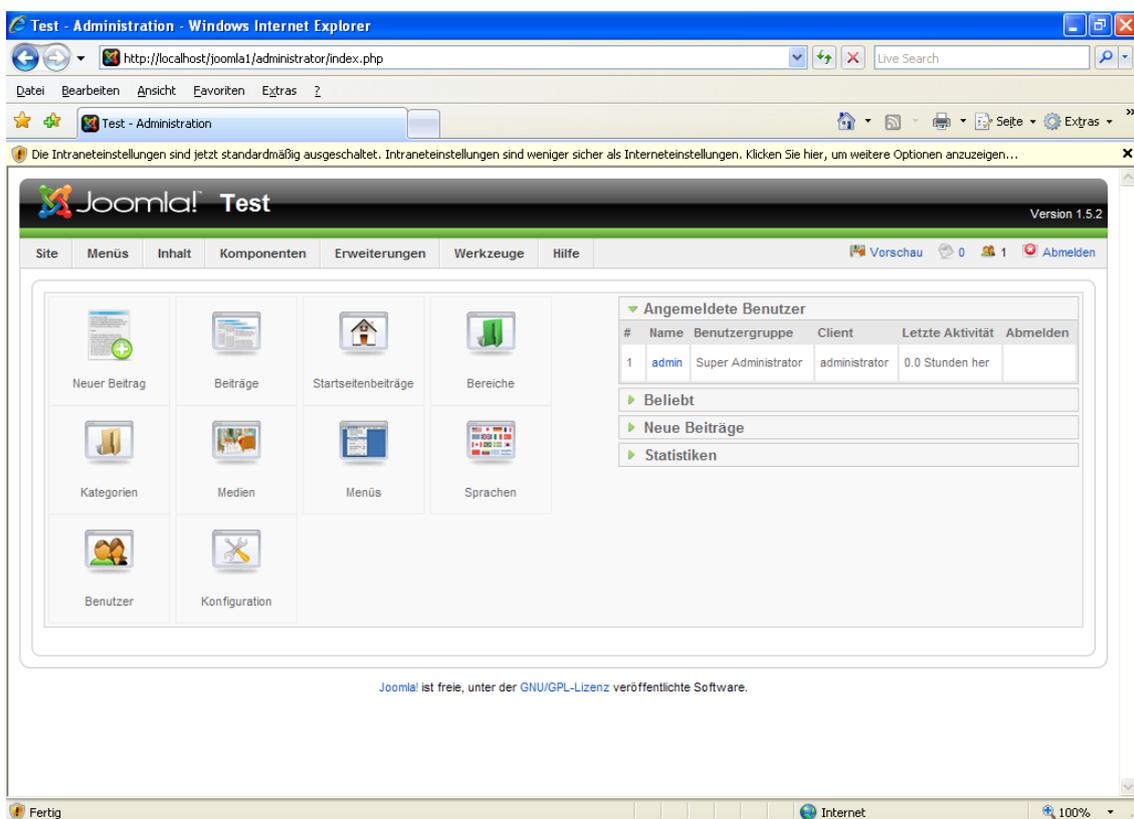


Abbildung 2: Backend eines frisch installierten Joomla!-Systems

Drupal

In Drupal ist die Administration in die eigentliche Webseite integriert. Ist man als Administrator bzw. berechtigter Benutzer eingeloggt, werden die zur Administration notwendigen Menüpunkte einfach im Hauptmenü zusätzlich zu den anderen Menüpunkten angezeigt. Die Anordnung des Administrationsbereiches hängt daher stark von der Struktur der Webseite ab. Im frisch installierten System ist die Administrationsoberfläche vom links befindlichen Hauptmenü durch einen Klick auf „Administer“ erreichbar. Die Oberfläche wird rechts im Inhaltsbereich der Webseite angezeigt.

Die Administrationsoberfläche von Drupal bietet sehr viele Funktionen an, wahlweise kann man sich dazu kurze Beschreibungen zu den einzelnen Menüpunkten einblenden lassen.

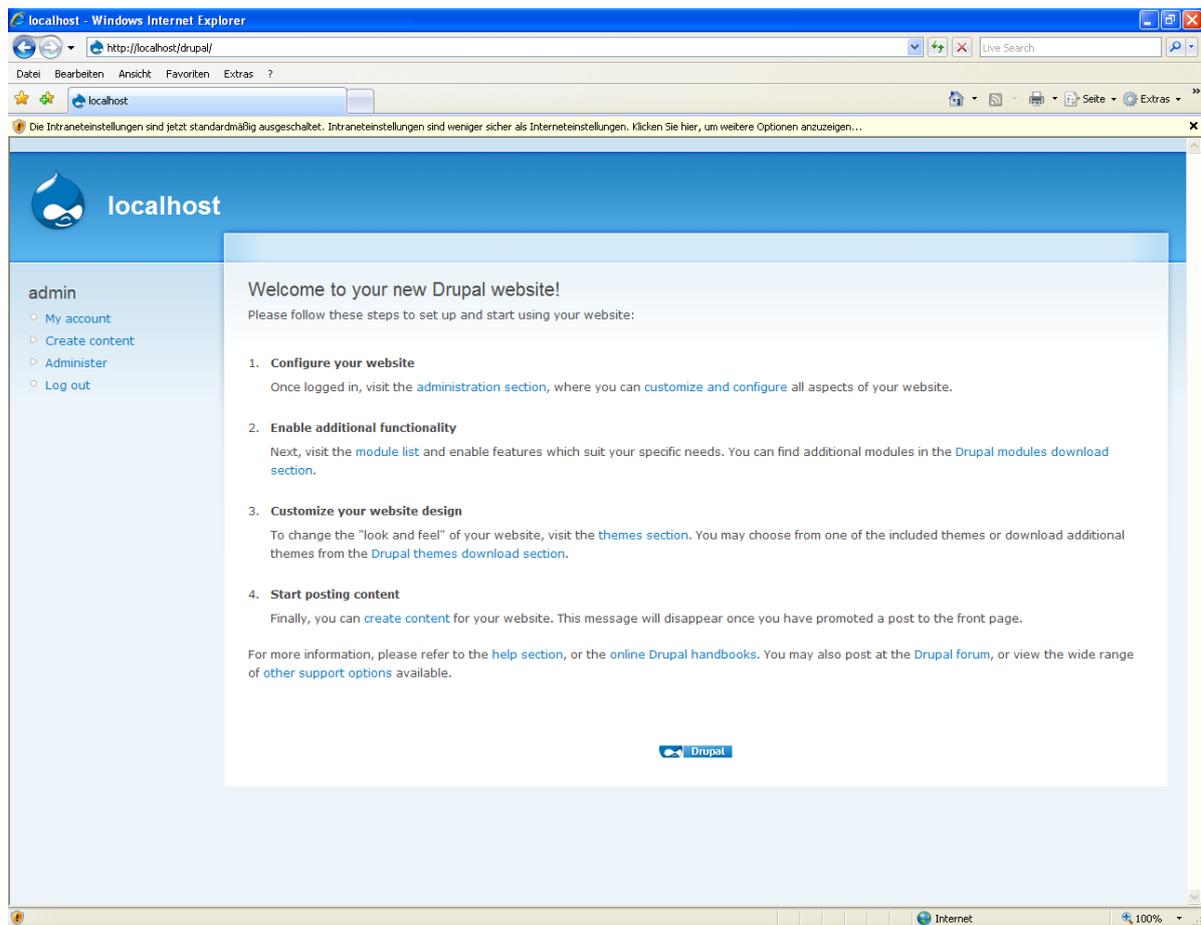


Abbildung 3: Die Oberfläche von Drupal

Auch hier gilt: Verschiedene Benutzer sehen nur jene Funktionen, die in ihrer Rolle für sie freigeschaltet sind.

3.3.2.3. Erstellung von Webseiten – Konzept

Typo3

Typo3 bietet als ausgereiftes Content Management System die Möglichkeit, die Inhalte einer Webseite getrennt von Struktur und Layout zu erstellen und zu verwalten. Im Folgenden werden die Möglichkeiten der Erstellung und Bearbeitung der verschiedenen oben angeführten Punkte vorgestellt.

Die grundsätzliche Struktur eines Webauftrittes, mit Seiten und Unterseiten wird in Typo3 durch einen Seitenbaum dargestellt, ähnlich der Ordnerstruktur in Windows-Betriebssystemen. Der Wurzelknoten enthält globale Einstellungen und Konfigurationen wie Backendbenutzer und Sprachen. Eine Ebene darunter liegt die eigentliche Root-Seite der Webseite, welche alle anderen Seiten als Unterseiten enthält. Diese Root-Seite ist gleichzeitig auch die Startseite des Webauftritts. Diese Art der Strukturierung und internen Darstellung

der Webseitenstruktur sorgt nicht nur für eine bessere Übersichtlichkeit, sondern ist auch bei der automatischen Erstellung der Navigation bzw. der Menüstrukturen von großer Bedeutung, da die Menüs der verschiedenen Ebenen aus dieser Struktur heraus generiert werden (können).

In Typo3 gibt es verschiedene Seitentypen für verschiedene Aufgaben. Neben den Standardseiten für inhaltliche Gestaltung gibt es noch andere, die auf externe bzw. interne Webpages verlinken. Spezielle Typen, keine Seiten im eigentlichen Sinn, sind der Papierkorb-Ordner und die „SysFolder“. Diese kann man sich im Seitenbaum als Ordner vorstellen, die als Container für bestimmte Elemente, wie Newsartikel oder auch Benutzer und Benutzergruppen dienen können. Prinzipiell können auch alle anderen Seiten nicht sichtbare Elemente, wie Newsartikel und Benutzer, aufnehmen. Aus Gründen der Übersichtlichkeit empfiehlt sich jedoch die Nutzung von SysFoldern.

Erstellt man neue Seiten, so sind diese leer. [SKÅ-04/3]

Die Struktur einer einzelnen Seite eines Webauftritts wird in Typo3 durch Vorlagen, sogenannte „Templates“, festgelegt.

Eine Typo3-Webseite benötigt mindestens ein „Root-Template“. Dieses muss der Stammseite der Webseite zugeordnet werden und enthält üblicherweise Informationen zur Struktur dieser Seite. Zusätzlich können noch weitere Informationen, wie Variablen, enthalten sein. Prinzipiell muss zwar jede Einzelseite ein Template besitzen, allerdings werden Templates von Seiten auf höheren Ebenen vererbt. Man benötigt daher für eine ganze Webseite mindestens ein Template.

Templates werden in der Typo3-eigenen Skriptsprache „TypoScript“ erstellt. Das Konzept von TypoScript verleiht Typo3 ungeheure Flexibilität. So wäre es einerseits möglich, eine Webseite komplett in einem Template mit Hilfe von TypoScript zu realisieren. Andererseits könnte durch einen einfachen Funktionsaufruf im Template die Erstellung der Seite komplett in eine PHP-Datei ausgelagert werden. Diese beiden Extreme sind für den Produktivbetrieb allerdings nicht praktikabel. In der Praxis führt man die Strukturierung einer Webseite in Typo3 mit HTML-Vorlagen durch. Dies sind statische HTML-Dokumente, deren HTML-Code die zukünftige Struktur der Seite enthält. Typo3 übernimmt die Struktur der Seite und füllt sie mit jenen Inhalten, die sich unformatiert in der hinter Typo3 liegenden Datenbank befinden. Der Prozess des „Füllens“ wird von der „TypoScript Frontend Engine“, kurz „TSFE“, erledigt.

Wo und wie Typo3 im statischen Dokument die dynamischen Inhalte einfügen bzw. ersetzen soll, kann auf verschiedene Weise konfiguriert werden. Folgend werden die zwei gängigsten Varianten kurz umrissen.

Die erste Möglichkeit, derzeit noch Standard in Typo3, besteht im Einsatz von „Markern“. Diese Marker müssen in der Form `###MARKERNAME###` bereits im statischen HTML-Dokument an den später zu ersetzenden Stellen eingefügt werden. Beim Einlesen („Parsen“) der HTML-Vorlage registriert die TSFE diese Marker und mit Hilfe einer Zuweisung mittels TypoScript können die Marker dann durch dynamischen Inhalt ersetzt werden. [SKÅ-04/1]

Die zweite Möglichkeit ist die Nutzung von „TemplaVoila“, kurz „TV“. Diese Erweiterung wurde von Kasper Skårhøj selbst entwickelt und mit Typo3 3.5 eingeführt. In aktuellen Versionen von Typo3 muss TV noch nachinstalliert werden, kommende Versionen haben TV bereits standardmäßig integriert.

TemplaVoila benötigt keine Marker, um in die statische HTML-Vorlage Inhalte einfügen zu können, denn in TemplaVoila können alle HTML-Tags im Dokument Marker sein. Über die TemplaVoila-eigene Oberfläche im Backend kann durch einfachen Klick auf den gewünschten HTML-Tag dieser mit Inhalten verknüpft werden. Zusätzlich können noch zahlreiche andere Einstellungen vorgenommen werden. TemplaVoila ist ein mächtiges Werkzeug, das die Erstellung von Webseiten wesentlich vereinfacht. [SKÅ-04/2]

Der Inhalt einer Seite wird in Typo3 grundsätzlich durch Seitenelemente, so genannten „page content elements“ strukturiert. Ein Seiteneditor ermöglicht die einfache Erstellung und Bearbeitung der page content elements, die Texte, Bilder, Links, Formulare, Tabellen, RSS-Feeds oder auch ganz eigene, selbstdefinierte Inhaltselemente sein können. Eine Seite kann aus vielen verschiedenen page content elements bestehen. Dies ermöglicht eine ungeahnte Flexibilität der inhaltlichen Gestaltung.

Standardmäßig bietet Typo3 vier unterschiedliche Typen von Inhaltselementen:

- *Typical page content*

Hier kann man Texte, Bilder, Listen und Tabellen in eine Seite einfügen. Typo3 bietet einen WYSIWYG-Editor, den sogenannten „RichTextEditor“ an, der es ermöglicht, Texte direkt im Backend zu formatieren. Hier wird zwar das Prinzip der Trennung von Layout, Inhalt und Struktur verletzt; dies erscheint in der praktischen Arbeit jedoch sehr sinnvoll und auch notwendig.

- *Special elements*

In dieser Kategorie werden neben Downloadlinks und Multimediaelementen wie Audio, Video oder Flash, auch die automatisierte Erstellung einer „Sitemap“ und die Möglichkeit, reinen HTML-Code in eine Seite einzubinden, angeboten.

- *Form elements*

Hier können grundlegende Formularanwendungen, wie ein einfaches Mail-Formular, eine Suchfunktion sowie ein Login-Formular erstellt werden.

- *Plugins*

Plugins werden in Typo3 jene Erweiterungen genannt, welche eigene Inhalte in die Webseite ausgeben. Typo3 bietet von vornherein vier Plugins für spezielle Anwendungen an. Es sind dies eine einfache Newsverwaltung, ein Diskussionsforum, ein schwarzes Brett sowie ein Gästebuch.[SKÅ-04/3]

Besitzen verschiedene Inhaltselemente immer die gleiche Struktur, wie dies z.B. in einem Produktkatalog der Fall ist, so kann man mit Hilfe von TemplaVoila eine Vorlage für diese Inhaltselemente erstellen. Diese mit TemplaVoila erstellten Inhaltselemente heißen „flexible content elements“.

Die Inhaltselemente werden einer Seite eindeutig zugeordnet. Um eine Seite erzeugen zu können, benötigt Typo3 die Information, wo die Inhaltselemente in der durch die Templates festgelegten Seitenstruktur eingebunden werden sollen. In der Standardinstallation stehen hier pro Seite vier verschiedene so genannte „Spalten“ zur Verfügung, welche mit Hilfe von TypoScript an vier verschiedenen Stellen der Seite integriert werden können. Jede Spalte kann beliebig viele „page content elements“ enthalten.

Weil die Limitierung auf vier Spalten pro Seite nicht sinnvoll erscheint, wurde mit TemplaVoila ein anderer Ansatz gewählt. Hier kann der Entwickler genau definieren, welche HTML-Tags der statischen HTML-Vorlage als Container für „page content elements“ genutzt werden sollen, die vier Spalten pro Seite werden dann durch die vom Entwickler definierten Container pro Seite ersetzt.

Zusätzlich können Inhalte auch direkt per TypoScript eingebunden werden. Dies ist vor allem bei manchen Erweiterungen notwendig, welche nicht als „page content elements“ auswählbar sind. Diese müssen dann direkt im Template der Seite einem bestimmten Marker zugewiesen werden.

Das Layout von Internetseiten wird heute vor allem mit Hilfe von CSS gestaltet. Diesem Umstand trägt auch Typo3 Rechnung. So ist es sehr einfach, eine externe CSS-Datei in eine Seite einzubinden. Für die in Typo3 vordefinierten Inhaltselemente können über die

Erweiterung „css_styled_content“ die Styleinformationen ebenfalls über ein XHTML-konformes Stylesheet bezogen werden.

Joomla!

Joomla trennt ebenso wie Typo3 die inhaltliche von der strukturellen Gestaltung, jedoch nach einem anderen Konzept.

Der inhaltliche Hauptbereich von Joomla! besteht aus sogenannten „Artikeln“. Diese bestehen aus Überschrift, Vorschautext und Haupttext.

Joomla! bietet ein Hierarchisierungssystem für Artikel an. Diese können einer frei definierbaren „Kategorie“ zugeordnet werden, welche wiederum einer frei definierbaren „Sektion“ zugeordnet sein muss. Nicht zuordenbare Artikel konnten bis Version 1.5 als sogenannter „static content“ angelegt werden. Seit Version 1.5 gibt es nun die vordefinierte Kategorie bzw. Sektion „Uncategorized“.

Die interne Struktur einer Webseite, wird in Joomla! komplett über die Menüverwaltung gesteuert. Joomla! erlaubt die Konfiguration von beliebig vielen Menüs, welche wiederum beliebig viele Menüpunkte haben können.

Bei der Erstellung eines Menüpunktes muss immer auch die Art der Seite angegeben werden, auf welche der Menüpunkt verweisen soll. Joomla! bietet hier einige vorgefertigte Arten von Seiten zur Auswahl an. So kann zum Beispiel ausgewählt werden, dass der Menüpunkt genau einen Artikel komplett anzeigen soll, oder alle Artikel einer bestimmten Kategorie, oder aber alle Artikel einer bestimmten Sektion in Blog-Form, oder auch, dass er auf die Startseite verweist.

Je nach gewählter Seitenart können dem Menüpunkt schließlich auch die Inhalte zugeordnet werden. Die zur Auswahl stehenden Seitenarten können auch durch Komponenten (Erweiterungen) ergänzt werden. So ermöglicht die Komponente „Web Links“ zum Beispiel in der Menükonfiguration die Auswahl der Seitenart „Weblinks“, welche Links zu externen Seiten anzeigt.

Jeder Inhalt einer Kategorie, einer Sektion, eines Menüs, allgemein eines Containers, kann in Joomla! gereiht werden. Dies ermöglicht es, die Reihenfolge der Anzeige von Artikeln zu ändern und auch die Reihenfolge der Menüpunkte in einem Menü, was vor allem für die Startseite relevant ist. In Joomla! ist immer der erstgereichte Menüpunkt des Hauptmenüs automatisch die Startseite.

Wie oben erwähnt, ist eine Seitenart auch „Startseite“, welche sich von anderen wählbaren Arten aber nur durch eine spezielle Anordnung der darauf befindlichen Inhalte unterscheidet.

Ob ein Artikel auf dieser Seite angezeigt wird, ist in Joomla! eine Eigenschaft des Artikels. Mit dem Frontpage-Manager können die „Startseiteninhalte“ dann sortiert werden. Die Seite des Inhaltstyps „Startseite“ muss nicht notwendigerweise die tatsächliche Hauptseite sein, da diese immer die verlinkte Seite des obersten Menüpunktes des Hauptmenüs ist – unabhängig von ihrem Seitentyp.

Die Struktur und das Layout einer Seite werden durch ein vorgefertigtes Template bestimmt. So ein Template besteht, anders als in Typo3, aus mehreren bestimmten Dateien, die in einer bestimmten Ordnerstruktur angelegt sein müssen.

Die Hauptdatei eines Templates wird in Joomla! über PHP eingelesen und verarbeitet, sie muss den Namen *index.php* tragen und im Verzeichnis *<joomla-Verzeichnis>/templates/<template_name>* liegen. In dieser Datei wird die Grundstruktur der Webseite vollständig festgelegt. Es muss bereits in der *index.php* festgelegt werden, wo die Inhalte und Module einer Seite später in das Template eingefügt werden. Zum Einfügen der dynamischen Elemente wird in Joomla! der Namensraum „JDOC“ verwendet.

Der folgende Code zeigt exemplarisch, wie die HTML-Vorlage gestaltet sein muss, um ein Modul einzubinden und den Inhaltsbereich festzulegen.

```
<div id="modul">
    <jdoc:include type="modules" name="top" style="xhtml" />
</div>
<div id="inhalt">
    <jdoc:include type="component" style="xhtml">
</div>
```

In den ersten DIV-Tag *modul* wird also das Modul mit dem Namen *top* eingebunden. Der Parameter *style=xhtml* bewirkt, dass das Modul von Joomla! in einem DIV-Element XHTML-konform ausgegeben wird. Andere mögliche *style*-Parameter sind *table*, durch welchen das Modul in Tabellenform ausgegeben wird; mit *none* würde es gar nicht formatiert werden.

Im anschließenden Tag *inhalt* wird die Inhaltskomponente eingebunden; hier werden die vom aktuellen Menüpunkt verlinkten Inhalte (Artikel, Blogbeiträge, etc.) angezeigt.

CSS-Dateien und eventuell benötigte Grafiken müssen in den Unterordnern *<template_name>/css* bzw. *<template_name>/images* gespeichert werden.

Ebenso benötigt wird die Metadatei *templateDetails.xml*, welche Informationen, wie Name des Templates, Kontaktdaten des Autors, die Versionsnummer, eine Beschreibung, sowie eine Liste aller zum Template gehörenden Dateien enthalten muss. Ein Vorschaubild zum Template, *template_thumbnail.png*, ist ebenfalls vonnöten.

Wird das so erstellte Template als ZIP-Archiv verpackt, so kann es weitergegeben und auf anderen Joomla!-Installationen automatisiert installiert werden.

Drupal

Auch Drupal sieht die Trennung zwischen Inhalten und Struktur/Layout vor, jedoch keine in Frontend und Backend. Die gesamte Organisation und Gestaltung von Drupal erfolgt über das Frontend. Das Administrationsmenü ist in die normale Menüstruktur der Seite eingebunden, die Menüpunkte werden jedoch nur für berechnigte, eingeloggte Benutzer angezeigt.

In der Organisation des Inhaltes bzw. auch in der Art des verfügbaren Inhaltes wird die Ausrichtung von Drupal auf Web 2.0-Anwendungen besonders klar ersichtlich.

Jeder Inhalt in Drupal, wie Foreneinträge, Artikel oder Blogeinträge, wird in einem sogenannten „Node“ gespeichert. Jeder Node besitzt eine eindeutige ID zur Identifikation und kann neben einigen anderen Eigenschaften wie Autor und Erstellungsdatum auch beliebig viele Kommentare besitzen. Ein Node kann über eine URL in der Form `<site-url>/node/<node-id>` eindeutig angesprochen und angezeigt werden.

Drupal besitzt standardmäßig folgende Nodetypen:

- *blog entry*
Eintrag im Drupal-eigenen Weblog.
- *book page*
Drupal ermöglicht die gemeinschaftliche Erstellung von Seiten im Rahmen eines „Buches“. Früher konnten nur der Nodetyp „book page“ in einem Buch verwendet werden, mittlerweile sind alle Nodetypen zulässig. Ein Buch besitzt sein eigenes Menü, das aus den Seitentiteln besteht. (vgl. Inhaltsverzeichnis)
- *comment*
Kommentare sind eigentlich keine eigenen Nodes, sie sind immer einer anderen Node zugeordnet. Konzeptuell werden sie trotzdem als Inhaltstyp verwendet.
- *forum*
Ein Forum ist eine geordnete Menge an Nodes, die dem gleichen Forumsnamen zugeordnet sind. Auch Foren selbst können hierarchisch organisiert sein.
- *page*
Die klassische statische Seite, die im Menü verlinkt wird.
- *poll*
Bietet die Möglichkeit, Abstimmungen durchzuführen.

- *story*

Generischer Seitentyp für Informationen, die sich im Laufe der Zeit ändern (z.B. Newseinträge). In der aktuellen Version von Drupal besteht praktisch kein Unterschied zwischen *story* und *page* mehr.

Nodes können durch frei definierbare Kategorien organisiert werden. Auch Kategorien selber sind schachtelbar.

Die Menüverwaltung in Drupal ist vielseitig. Es können beliebig viele Menüs erstellt werden. Bestimmte Menüpunkte sind auf Grund des Administrationskonzepts bereits ab dem Zeitpunkt der Installation vorhanden, jedoch nur dem (eingeloggten) Administrator zugänglich. Die Menüeinträge können frei definiert werden. Die Hierarchie der Menüpunkte, auch der vorgegebenen, ist beliebig änderbar. Ein Menüpunkt verweist entweder auf Nodes (über interne URLs, z.B. *node/<node-nr>*) oder auf eine externe URL oder über Kürzel auf spezielle Seiten (so verlinkt z.B. *<front>* zur Startseite).

Es ist auch möglich, das Menü auf Basis der Kategorien generieren zu lassen. [DRC-07]

Inhalte, die außerhalb des definierten Inhaltsbereichs, etwa in der linken oder rechten Spalte, bzw. in der Kopf- und Fußzeile angezeigt werden sollen, sind „Blocks“ organisiert. Blocks können statisch oder dynamisch sein. Beispiele für statische Blocks sind der Login-Block und der Menü-Block. Ein sich selbst aktualisierendes RSS-Anzeigemodul ist beispielsweise ein dynamischer Block.

Die Struktur und das Layout einer Seite werden in Drupal durch sogenannte „Themes“ realisiert.

Ähnlich wie in Joomla! sind Themes eine Sammlung von Dateien. Im Unterschied zu Joomla! ist in Drupal eine zusätzliche Abstraktionsschicht eingezogen – die „theme engine“ mit optionaler „tagging language“. Die theme engine übersetzt die Informationen der Theme-Dateien in einen von Drupal verwendbaren Code. Der große Vorteil dieses Konzepts ist, dass Drupal mit jeder Templating-Sprache arbeiten kann, für die eine theme engine existiert. Als Standard wird in Drupal „*phptemplate*“ verwendet. Ähnlich wie in Joomla! bindet diese engine über PHP-Befehle in den theme-Dateien die Inhalte von Drupal in die statische HTML-Vorlage ein.

Jedes Theme muss eine *.info* – Datei besitzen. Sie enthält Metainformationen zum Template, Die einzigen beiden Informationen, die zwingend benötigt werden, sind Name und jene Version von Drupal, ab der das Template eingesetzt werden kann. Optional ist die Eingabe von Beschreibung, Versionsnummer sowie benötigter theme engine. Ebenso können benötigte JavaScript- und CSS-Dateien angegeben werden. Das *.info*-File ermöglicht auch die Angabe

der unterstützten Features. Wenn also beispielsweise ein Theme kein Suchfeld unterstützt, kann das im *.info*-File auch eingetragen werden. Die Definition der vom Template unterstützten Inhaltselemente (für den Hauptinhalt und die Blockelemente) ist ebenfalls möglich. Werden Features und Inhaltselemente nicht angegeben, nimmt Drupal Standard-Werte an. Der Name der *.info*-Datei ist gleichzeitig der maschinenlesbare Name des Themes. Die eigentlichen Template-Dateien enthalten jeweils das HTML-Gerüst für ein Inhaltselement, wie „page“, „block“, „node“ oder „comment“. Sie heißen auch genauso wie die von ihnen abgebildete Region, also z.B. *page.tpl.php* oder auch *block.tpl.php*. Durch diese Modularisierung können die Template-Dateien meist sehr klein und übersichtlich gehalten werden. In die Dateien müssen auch die Befehle zum Einbinden der Drupal-Inhalte eingebettet werden. Je nach gewählter theme-engine sehen diese natürlich anders aus. Im Standardfall phptemplate werden PHP-Befehle benutzt, die mit Drupal-Variablen arbeiten. So wird in der folgenden Zeile in *block.tpl.php* des „garland“-themes das Subject des aktuell einzufügenden Blocks ausgegeben.

```
<h2><?php print $block->subject ?></h2>
```

Ist in einem Theme Programmlogik notwendig, etwa um die von Drupal gelieferten Variablen vor der Ausgabe zu verändern, so kann dies in der (optionalen) Datei *template.php* geschehen. Ein Theme kann neben den benötigten CSS- und Scriptdateien noch ein Vorschaubild sowie ein Logo enthalten.

Ein besonderes Konzept ist die Theme-Vererbung. In der *.info*-Datei eines Themes kann ein „base theme“ angegeben werden, auf welchem das aktuelle Theme basieren soll. Im aktuellen Theme müssen dann nur jene Teile angepasst werden, die sich vom base theme unterscheiden sollen – die anderen Teile werden einfach geerbt. Die Theme-Vererbung kann in beliebiger Tiefe erfolgen. [DRU-08/3]

Fazit

Die Trennung zwischen Inhalten und Layout/Struktur ist für ein gutes Content Management System ein Muss. Alle getesteten CMS unterstützen dieses Feature.

Vom Templating-System besitzt Typo3 mit TemplaVoila sicher das durchdachtste Konzept. Im Gegensatz zu Drupal und Joomla! werden in das System wirklich nur HTML-Dateien eingebunden, was die Erstellung und nachträgliche Änderbarkeit auch ohne Kenntnisse von Typo3 sehr erleichtert. Die Zuordnung von Inhalten zu HTML-Tags erfolgt in TemplaVoila durch einfaches Klicken, ganz ohne Programmiersprache.

Die vordefinierten Inhaltstypen aller drei CMS scheinen für eine allgemeine Seite sehr sinnvoll, wobei bei Drupal deutlich die Ausrichtung auf Web 2.0-Anwendungen bzw. auf

„community-driven Websites“ erkennbar ist. Typo3 bietet hier wiederum den Vorteil, dass sich neue Inhaltselemente durch TemplaVoila sehr einfach realisieren lassen.

Für die Anwendung Portal „Netzwerke und Sicherheit“ sind aus konzeptueller Sicht der Webseitenerstellung alle drei getesteten Systeme geeignet.

3.3.2.4. Erweiterungskonzept

Typo3

Erweiterungen werden in Typo3 als „Extensions“ bezeichnet. Sie können über den „Extension Manager“ installiert und verwaltet werden. Dabei existieren zwei grundsätzliche Arten von Extensions: Jene Extensions, welche Funktionalität zum Frontend hinzufügen, werden „Plugins“ genannt. Extensions im Backend heißen „Module“.

Durch Typo3-Extensions können viele weitere Inhaltselemente erstellt werden. Das vorhin erwähnte TemplaVoila ermöglicht zum Beispiel die Erstellung von Vorlagen für wiederkehrende Inhalte, so genannte „flexible content elements“, welche dann ebenfalls ein „page content element“ sein können. Zu erwähnen ist noch, dass es eine Extension gibt, welche die Erstellung von eigenen Extensions unterstützt und sehr vereinfacht.

Drupal

Erweiterungen heißen in Drupal „Module“. Diese können sowohl neue Node- als auch Blocktypen zur Verfügung stellen bzw. eigene Menüpunkte zur Menüleiste hinzufügen. Ein Modul ist beispielsweise das „Content Construction Kit“, welches dem Benutzer die Erstellung von eigenen Nodetypen ermöglicht.

Derzeit (Stand: 3. Juni 2008) sind rund 3900 Module in diversen Kategorien wie Sicherheit, E-commerce oder Administration auf drupal.org verfügbar. [DRU-08/2]

Joomla!

Neben den eigentlichen Inhalten gibt es in Joomla! noch Module, Komponenten und Plugins (ab Version 1.1, vorher hießen die Plugins „Mambots“).

Module dienen der Anzeige von Daten, die keine Inhalte im obigen Sinn sind. Diese Daten können z.B. von Komponenten oder von Joomla! selbst stammen. Module können Parameter besitzen und konfigurierbar sein, so kann etwa das Menü-Modul ein Menü horizontal, vertikal oder als formatierbare HTML-Liste ausgeben. Jedes Modul besitzt einen eindeutigen Namen, was bei der Erstellung von Templates notwendig ist, um Module in eine Vorlage einbinden zu können. Beispiele sind die Suchfunktion, das Login-Modul, das Statistikmodul und eben das Menü-Modul.

Plugins sind kleine Erweiterungen, die direkt in die Programmlogik vom Joomla! eingreifen. Sie ermöglichen es, Inhalte vor der Ausgabe zu verändern. Beispielsweise werden durch das Plugin „Email Cloaking“ alle angezeigten Email-Adressen vor der Ausgabe mit Hilfe von JavaScript codiert, damit sie von Crackern nicht automatisiert ausgelesen und für Werbezwecke missbraucht werden können.

Komponenten sind die klassischen Erweiterungen in Joomla!. Sie können eigene Daten verwalten, können im Backend eigens konfiguriert werden und besitzen auch Zugriff auf die Datenbank. Komponenten werden meist im Inhaltsbereich angezeigt. Manche Komponenten besitzen zur Interaktion mit dem Benutzer eigene Module. Ein Beispiel dafür ist ein Gästebuchsystem. Auch das Poll-System ist eine Komponente.

Derzeit (Stand 2. Juli 2008) sind ca. 3900 Erweiterungen für Joomla! verfügbar.

[EBE-07, GRA-07]

3.3.2.5. Features des Redaktionssystems

Typo3

- *WYSIWYG-Editoren*

Typo3 ermöglicht die Formatierung von Inhalten mit Hilfe eines RichText-Editors, der nach dem WYSIWYG-Prinzip funktioniert. So können neben den diversen Schriftattributen auch Listen, Tabellen, Links, Sonderzeichen und Einzüge erstellt werden. Daneben bietet der Editor auch eine „Rückgängig-Funktion“ an.

Zusätzlich gibt es auch für andere Inhaltselemente, wie zum Beispiel Formulare, Editoren.

- *Editieren von Inhalten im Frontend*

Typo3 bietet eingeloggten Backendbenutzern die Möglichkeit, Inhalte direkt im Frontend zu editieren, was vor allem unerfahrenen Redakteuren die Arbeit erleichtern soll. Auch hier kann der RichText-Editor benutzt werden, das Ergebnis der Änderung ist auf der Seite sofort sichtbar.

- *Zeitgesteuerte Veröffentlichung und Archivierung von Inhalten*

Jede Seite und jedes Inhaltselement kann auf die Minute genau veröffentlicht und wieder vom Netz genommen werden. So können auch am Wochenende oder bei Abwesenheit des Redakteurs neue, aktuelle Inhalte gebracht werden bzw. alte Inhalte automatisch vom Netz genommen werden.

- *Anpassung des Backends für den einzelnen Benutzer*

Durch die Rechteverwaltung in Typo3 ist es möglich, die Zugriffsmöglichkeiten eines Benutzers auf die von ihm benötigten Funktionen zu beschränken. Das hat zwei

Vorteile: Einerseits stellt es eine Erhöhung der Sicherheit dar, da ein Redakteur abseits seines Zuständigkeitsbereichs weder absichtlich noch unabsichtlich etwas an der Seite ändern kann; andererseits sorgt es auch für bessere Übersichtlichkeit und damit leichtere Einarbeitung für den Redakteur.

- *Automatische Verarbeitung von Bildern*

Typo3 ermöglicht die Einbindung von Grafikbearbeitungsbibliotheken wie ImageMagick oder GraphicsMagick. Damit können in Typo3 eingebettete Bilder manipuliert und dynamisch bearbeitet werden. Für jedes hochgeladene Bild wird automatisch ein Vorschaubild, ein so genanntes „Thumbnail“, erzeugt. Mit Hilfe dieser Bibliotheken lassen sich auch Grafiken aus Inhalten dynamisch generieren, z.B. für Überschriften oder Menüpunkte

- *Metadaten*

Typo3 ermöglicht die Eingabe von zahlreichen, durchsuchbaren Metadaten zu Seiten. Autor, Erstellungsdatum und ähnliche Informationen können automatisch gesetzt, aber auch durch den Redakteur verändert werden.

Joomla!

- *WYSIWYG-Editor*

Auch Joomla! bietet einen WYSIWYG-Editor zur Veränderung des Inhaltes, den „TinyMCE 2.0“. Dieser bietet neben den Standards wie Texteigenschaften und Tabellen, Links, Einzüge, Listen, usw. noch die Möglichkeit, Medien direkt in den Text einzubetten.

- *Editieren von Inhalten im Frontend*

Joomla! bietet ebenfalls die Möglichkeit, Inhalte im Frontend mit Hilfe des Editors „TinyMCE 2.0“ zu bearbeiten. Der Benutzer benötigt dafür mindestens die Rechte eines „Editors“, muss aber kein Backendbenutzer sein.

- *Zeitsteuerung von Inhalten*

Jeder Inhalt in Joomla! kann zeitgesteuert veröffentlicht und wieder vom Netz genommen werden, theoretisch sogar auf die Sekunde genau. Auch das Erstellungsdatum und der Autor können manuell festgelegt werden.

- *Metadaten*

Für jeden Beitrag können Metadaten, wie Beschreibung und Schlüsselwörter definiert werden. Das soll unter anderem die Indizierung einer Seite für Suchmaschinen erleichtern.

- *Export als PDF*

In Joomla! kann jeder Artikel so konfiguriert werden, dass er als PDF exportiert werden kann. Über ein Icon neben der Überschrift des Artikels kann dann der Artikel als PDF angezeigt werden.

- *Druckansicht*

Ebenso ist es möglich, jeden Artikel automatisch in eine Druckansicht bringen zu lassen.

- *Artikellink per Mailformular versenden*

Ein Link zu jedem Artikel kann per Formular via Email versandt werden. Dies dient etwa dazu, einen Freund auf einen Artikel hinzuweisen.

- *Suchmaschinenfreundliche URLs*

Joomla! ermöglicht die Verwendung von suchmaschinenfreundlichen URLs, welche auch für den Menschen besser zu lesen sind. Diese präsentieren die Seite mit scheinbar statischen URLs, wobei die Seite trotzdem im Hintergrund dynamisch aus der Datenbank generiert wird.

- *Bildverarbeitung*

Joomla! verwendet zur Bildbe- und verarbeitung das „GD Toolkit“, welches jedoch erst im Zusammenspiel mit anderen Komponenten benutzt wird.

- *Übernahme von Newsfeeds*

Joomla! bietet in einer ab dem Zeitpunkt der Installation enthaltenen Komponente die Möglichkeit, Newsfeeds von fremden Webseiten in die eigene zu integrieren.

Drupal

- *Suchmaschinenfreundliche URLs*

Auch Drupal ermöglicht die Nutzung von suchmaschinenfreundlichen URLs.

- *Automatische Übernahme und Produktion von Newsfeeds*

Beliebige RSS- und Atom-Feeds können von Drupal verarbeitet und in die eigene Webseite integriert werden. So können ohne Administrationsaufwand Inhalte dynamisch aktualisiert werden.

Inhalte können auch dynamisch als RSS- bzw. Atom-Feeds angeboten werden.

- *Ausgereiftes Kommentarsystem*

Für jede Node, also jegliche Form von Inhalt kann in Drupal eine Kommentarfunktion aktiviert werden, welche es Besuchern ermöglicht, ihre Meinung kundzutun.

- *Bildverarbeitung*

Auch Drupal verwendet zur Bildbe- und -verarbeitung das „GD Toolkit“. Bildbearbeitung und -verarbeitung sind allerdings erst in Zusammenhang mit anderen Modulen interessant.

Fazit

Typo3 und Joomla! besitzen ab Installation bereits viele nützliche Funktionen wie WYSIWYG-Editoren und die zeitgesteuerte Veröffentlichung und Archivierung von Inhalten. Joomla! sticht besonders durch die Exportfunktion für PDF und die Druckansicht hervor, während Typo3 eine besonders leistungsstarke Bildverarbeitung bietet. Großer Vorteil von Typo3 ist auch, dass durch die Rechteverwaltung jeder Benutzer nur jene Teile des Backends sieht, die er tatsächlich benötigt – was zu einer enormen Vereinfachung des Bearbeitungsvorganges führt. Der Schwerpunkt von Drupal mit dem Kommentarsystem und der einfachen Einbindung und Generierung von Newsfeeds liegt auch hier eindeutig auf Web 2.0 Anwendungen. Drupal bietet im Installationszustand etwas weniger Funktionalität. Funktionen wie der WYSIWYG-Editor oder die Zeitsteuerung lassen sich jedoch mit Modulen nachrüsten.

Auch in diesem Punkt gilt wieder: jedes der Systeme wäre für das Projekt „Security-Portal“ grundsätzlich geeignet.

3.3.3. Sicherheit

3.3.3.1. Rechtesystem/Benutzermodell

Typo3

Typo3 unterscheidet zwischen „Frontend-“ und „Backendbenutzern“. Zusätzlich gibt es jeweils eigene Benutzergruppen. Frontendbenutzer können nur auf Frontendinhalte, also nur auf die eigentliche Webseite zugreifen, dies dient zur Einrichtung von internen, geschützten Bereichen auf einer Webseite. Die Zugriffseinstellungen können für jede Frontendbenutzergruppe pro Seite konfiguriert werden. Frontendbenutzer und -gruppen werden in einem „SysFolder“ gespeichert, der sich an beliebiger Stelle im Seitenbaum befinden kann. Im Seitentemplate der Root-Seite muss mit Hilfe von TypoScript auf den SysFolder verwiesen werden. Damit sich Benutzer selbst registrieren können, muss eine Erweiterung installiert werden. Standardmäßig wird dieses Feature nicht angeboten.

Backendbenutzer können sich sowohl im Front- als auch im Backend einloggen. Diese Benutzer bzw. die Benutzergruppen müssen in der Wurzel des Seitenbaumes angelegt werden.

Für jeden Backendbenutzer und jede -gruppe kann explizit angegeben werden, welche Backendmodule sichtbar sein sollen und welche Datenbanktabellen gelesen oder verändert werden dürfen. Ebenso können die verfügbaren Seiten- und Inhaltstypen eingeschränkt werden. Jede Eigenschaft einer Seite, wie auch der Inhaltstypen, kann einzeln aktiviert werden. Ist eine Seite mehrsprachig, so kann die für einen Benutzer bearbeitbare Sprache konfiguriert werden, ebenso jene Seite im Seitenbaum, die der User als Ausgangspunkt im Seitenbaum sehen soll. Gleiches gilt auch für die Ordner im Dateisystem. Weiters kann limitiert werden, ob der Benutzer „Live-“ und/oder „Draft-Workspace“ bearbeiten kann und ob er neue Projekte anlegen darf.

Für sämtliche Benutzer(gruppen) des Backends und Frontends ist zusätzlich die Limitierung des Zugriffs auf einen bestimmten Bereich von IP-Adressen möglich. So kann verhindert werden, dass sich ein Redakteur z.B. von außerhalb eines Büros einloggt.

Joomla!

Das Rechtesystem in Joomla! bietet sieben verschiedene Benutzergruppen an, welche noch einmal in Frontend-Benutzer und Backend-Benutzer unterteilt sind. Es ist explizit ab Installation nicht möglich, eigene Benutzergruppen oder Rollen zu erstellen. Es gibt zwar eine Komponente, die eine erweiterte Benutzerverwaltung möglich macht, die Installation derselben überschreibt jedoch einige Core-Dateien von Joomla! und ist daher im Sinne der Wartbarkeit nicht zu empfehlen.

Jeder angelegte Frontend- und Backendbenutzer muss einer der sieben Benutzergruppen angehören.

- *Frontend-Benutzergruppen:*

Frontend-Benutzer können sich nur im Frontend einloggen. Auf das Backend haben sie keinen Zugriff. Es gibt folgende Benutzergruppen:

- *Registered*

Dieser Benutzer ist registriert und darf Teile der Webseite sehen, die ein nicht eingeloggtter Benutzer nicht sieht.

- *Author*

Wie „Registered“, zusätzlich darf dieser Benutzer Inhalte verfassen (z.B. Newsartikel) und seine eigenen Artikel auch ändern – es ist jedoch nicht

möglich, dem Benutzer nur das Verfassen von Inhalten in bestimmten Bereichen zu erlauben – diese Erlaubnis gilt auf der ganzen Webseite. Die von ihm verfassten Inhalte müssen von einem Mitglied der Benutzergruppe Publisher oder höher freigeschaltet werden.

- *Editor*

Wie „Author“, zusätzlich darf er auch Inhalte von anderen Benutzern verändern, jedoch keine Inhalte veröffentlichen.

- *Publisher*

Wie „Editor“, zusätzlich darf er Inhalte veröffentlichen.

- *Backend-Benutzergruppen:*

Jeder Backend-Benutzer ist immer auch ein Frontend-Benutzer. Alle Backend-Benutzer sind Mitglied der Gruppe „Special-User“.

Es werden folgende Backend-Benutzergruppen unterschieden:

- *Manager*

Ein „Manager“ darf verschiedene Informationen über die Seite im Backend sehen und Inhalte auch im Backend erzeugen.

Er darf weder Benutzer verwalten, noch Module und Komponenten installieren, noch den Menüpunkt Site-Konfiguration ändern, noch Rundmails versenden und auch keine Templates und Sprachdateien ändern und installieren.

- *Administrator*

Ein „Administrator“ darf alles, außer „Super-Administratoren“ erstellen und bearbeiten, den Menüpunkt Site-Konfiguration ändern, ein Rundmail an alle Benutzer verschicken und Templates und Sprachdateien ändern und installieren.

- *Super-Administrator*

Darf alle Funktionen von Joomla! bedienen. [GRA-06]

Drupal

Drupal verwendet ein Rollenmodell zur Rechte- und Benutzerverwaltung. Im Installationszustand hat Drupal genau einen Benutzer, welcher alle Rechte besitzt, den „Administrator“. Ebenso sind ab der Installation zwei Rollen vorhanden, „anonymous user“ und „authenticated user“. Anonymous user ist die Rolle, welche alle nicht-eingeloggten Besucher einer Webseite zugewiesen bekommen. Authenticated user sind alle registrierten

und eingeloggten Besucher. Ausnahmslos jeder Besucher einer Drupal-verwalteten Seite bekommt automatisch eine Rolle zugewiesen.

In Drupal ist es ab der Installation möglich, dass sich Seitenbenutzer über das Frontend selbst registrieren – wahlweise mit bzw. ohne Freischaltung durch den Administrator. Drupal bietet auch die Möglichkeit, die Gültigkeit der Email-Adresse durch das Zuschicken eines Passworts zu verifizieren, sowie die Möglichkeit bei Löschung, Deaktivierung und zum Zurücksetzen des Passworts automatisierte, vorkonfigurierte Emails zu verschicken.

Jeder registrierte Benutzer kann neben der Rolle authenticated user noch eine unbegrenzte Zahl an weiteren Rollen besitzen.

In Drupal kann man beliebig viele Rollen definieren. Die Rechte, welche eine Rolle besitzen kann, werden von den Modulen „angeboten“. So bietet das “comment module” die Rechte “access comments”, “administer comments”, “post comments” und “post comments without approval”. Im neu installierten Drupal können so die Rechte von sieben Modulen administriert werden.

Zusätzlich bietet Drupal noch eine Filterfunktion an, sodass Benutzernamen, Email-Adressen oder der Zugriff von bestimmten Hosts explizit erlaubt bzw. explizit verboten sind. Die Verwendung von Wildcards ist möglich.

Fazit

Typo3 ist in der Kategorie „Benutzer- und Rechteverwaltung“ im Sinne der Sicherheit und auch der Möglichkeiten klarer Favorit. Eine derartig feingranulare Rechteverwaltung bietet sonst keines der getesteten Content Management Systeme.

Drupal bietet mit dem implementierten Rollenmodell ein durchdachtes Sicherheitssystem an, welches in der Granularität aber nicht an Typo3 heranreicht. Was die Sicherheit betrifft, ist die Aufhebung der Trennung zwischen Administration des Webangebots und der eigentlichen Webseite ein konzeptueller Nachteil.

Expliziter Nachteil von Joomla! ist die Limitierung auf sieben Benutzergruppen ohne weitreichende Konfigurationsmöglichkeiten.

3.3.3.2. Umgang mit Sicherheitslücken

Typo3

Wie in der Einleitung bereits erwähnt, ist ein eigenes „Security-Team“ von acht Personen für die Behebung von Sicherheitslücken in Typo3 verantwortlich. Dabei wird nicht nur der Systemkern von Typo3, also jene Komponenten in der Standardinstallation auf

Sicherheitslücken überprüft, sondern auch in den Erweiterungen im „Typo3 Extension Repository“, kurz „TER“ werden diese gesucht und ausgebessert.

Dabei verfolgt das Team den Ansatz der „least disclosure“. Das bedeutet, dass eine Sicherheitslücke erst an die Öffentlichkeit getragen wird, wenn eine Korrektur, ein „Fix“ dafür vorhanden ist.

Bei Sicherheitsproblemen in einer Extension wird zwar der Autor mit der Bitte um Korrektur kontaktiert, das Security-Team steht ihm jedoch bei Bedarf zur Seite.

Ist eine Sicherheitslücke durch Herausgabe eines Patches gefixt, so wird ein „Security-Bulletin“ veröffentlicht, mit genauer Beschreibung des Problems sowie Anweisungen zur Behebung.

Im Jahr 2008 wurden bis zum 16. Mai neun Security-Bulletins veröffentlicht, wobei keines der darin beschriebenen Probleme den Typo3-Core betroffen hat.

Das Security-Team hat als Leitfaden für Entwickler das „Typo3 Security Cookbook“ herausgegeben. Das Dokument enthält Ratschläge zu sicheren Konfiguration von Typo3, MySQL und PHP sowie dem verwendeten Webserver.[TYP-07/4]

Drupal

In Drupal ist der Umgang mit Sicherheitslücken ähnlich organisiert wie in Typo3. Auch hier gibt es ein eigenes „Security-Team“, bestehend aus 28 Personen, welches neben der aktiven Suche nach „Vulnerabilities“ im Drupal-Core sowie in den Modulen, bekannt gewordene Lücken schnellst möglich ausbessert, bzw. den Entwickler des betroffenen Modules darum bittet. Auch das Drupal-Security-Team gibt erst nach dem Fix einer Sicherheitslücke diese in einem „security announcement“ öffentlich bekannt, um die Ausnutzung von nicht behobenen Vulnerabilities nicht zu fördern. Im Jahr 2008 wurden bis 19. Mai 30 security announcements herausgegeben, fünf davon betrafen den Drupal-Core.

Für Entwickler wird eine Dokumentation „Writing Secure Code“ angeboten. In dieser werden die hauptsächlichen Angriffsvektoren gegen Drupal-Systeme identifiziert, unter anderem „Cross site scripting“ und „SQL injection“, sowie Tipps zur Prävention gegeben. [DRU-08/4, DRU-08/5]

Joomla!

Auch in Joomla! gibt es ein Security Team, welches aus nicht näher genannten „hoch vertrauenswürdigen, langjährigen Joomla! Entwicklern“ besteht. Ihre Vorgehensweise gleicht den Teams bei Drupal und Typo3. Nach der Meldung wird die Sicherheitslücke gefixt und

erst gemeinsam mit einem Patch oder einem neuen Release von Joomla! veröffentlicht. [JOO-08/3]

Für die Behebung von Sicherheitslücken in Zwischenversionen, so genannten „maintenance versions“ ist jedoch die „Development Working Group“ verantwortlich, welche auch für die grundsätzliche Weiterentwicklung zuständig ist. [JOO-08/4]

Es existieren auch zahlreiche Dokumente mit Anleitungen, wie Joomla! und die verwendeten Technologien konfiguriert werden sollen, um möglichst sicher zu sein. Diese sind jedoch nicht zentral verfügbar.

Fazit:

Typo3 und Drupal nehmen Sicherheitsbelange sehr ernst, bieten eigene Bereiche auf ihren offiziellen Webseiten mit Informationen zur Sicherheit, zum Prozess der Behebung von Sicherheitslücken und zu den verantwortlichen Teams. Drupal und Typo3 sind in diesem Punkt als gleichwertig zu betrachten.

In Joomla! gibt es zwar auch ein eigenes Sicherheitsteam, allerdings widerspricht sich die offizielle Webseite in diesem Punkt an anderer Stelle selbst. Großer Nachteil ist auch, dass die Informationen, wie man Sicherheitslücken melden soll, sehr versteckt sind und es keinen zentralen Bereich gibt, in dem alle, die Sicherheit betreffenden, Fragen organisiert sind.

Auch gibt es keine Security Announcements, sodass bestenfalls über die „Feature List“ eines neuen Releases herausgefunden werden kann, ob und welche Sicherheitslücken damit behoben werden.

3.3.3.3. CMS-interne Sicherheitskonzepte

Typo3

Typo3 bietet zur Erhöhung der Sicherheit viele Möglichkeiten, welche über das Install-Tool konfiguriert werden können. Dieses ist passwortgeschützt, zusätzlich kann es durch Löschen des *typo3/install* – Verzeichnisses komplett deaktiviert werden.

Eine Auswahl der von Typo3 angebotenen Sicherheitsfeatures:

- *Limitierung von Entwicklungsfehlermeldungen auf bestimmte IP-Adressen*

Fehlermeldungen sollen eine möglichst genaue Fehlerlokalisierung ermöglichen, um die Entwicklung von Systemen zu erleichtern. Kennt man die Fehlermeldungen eines Systems, so findet man möglicherweise leichter Angriffspunkte. Die Kontrolle der Ausgabe von Fehlermeldungen ist daher ein wichtiges Feature.

- *Restriktion der hochladbaren bzw. manipulierbaren Dateitypen*

Typo3 ermöglicht im Backend das Hochladen und das Bearbeiten von Dateien am Webserver. Zur Erhöhung der Sicherheit können bestimmte Dateitypen von diesen Aktionen ausgeschlossen werden. So dürfen im Installationszustand z.B. keine PHP-Dateien hochgeladen werden.

- *Sicherheit bei Extensions*

Ein großes Sicherheitsproblem stellen für gewöhnlich die Erweiterungen dar. Jedermann kann Extensions erstellen und anderen Benutzern zur Verfügung stellen. Bestimmte Extensions werden von Typo3-Sicherheitsexperten begutachtet und wenn sie als sicher eingestuft werden, mit dem Prädikat „Reviewed“ ausgezeichnet. Obwohl dies keine 100-prozentige Sicherheit gewährleistet, können „reviewed“ Extensions relativ bedenkenlos eingesetzt werden.

Der Umgang mit Extensions im System selbst kann ebenfalls limitiert werden. So kann die Möglichkeit der Bearbeitung von Extension-Dateien im System deaktiviert werden, weiters kann die Installation von zusätzlichen Extensions unterbunden werden.

- *Logging*

Typo3 bietet eine Vielfalt an Möglichkeiten des Loggings. Sicherheitsrelevant sind hier vor allem erfolgreiche und fehlgeschlagene Login-Versuche. Diese werden automatisch mit Datum, Zeit und IP-Adresse festgehalten. Zusätzlich kann bei vier fehlgeschlagenen Login-Versuchen pro Stunde eine Benachrichtigungsemail geschickt werden. Alternativ kann bei jedem Backend-Login ein Email geschickt werden.

- *Zugriffskonfiguration*

Jeder Backend- und Frontendzugriff kann zusätzlich auf bestimmte IP-Adressen limitiert werden. So wird sichergestellt, dass z.B. niemand außerhalb des Firmennetzwerkes die Login-Seite des Backends aufrufen kann.

Einstellbar ist auch, dass „Sessions“ sowohl im Frontend als auch im Backend auf ihre IP-Adressen gelockt werden. Dies ist eine Sicherheitsmaßnahme gegen „Session-Hijacking“, führt bei Neuvergabe von DHCP-vergebenen IP-Adressen jedoch zur Notwendigkeit eines erneuten Einloggens.

Die gesamte Kommunikation mit dem Backend kann auf SSL-Basis umgestellt werden (HTTPS), alternativ dazu kann auch nur der Login-Vorgang verschlüsselt abgewickelt werden.

Damit auch ein Administrator keinen Zugriff auf die Passwörter hat, werden nur die Hash-Werte der Kennwörter in der Datenbank gespeichert. Beim Login-Vorgang wird zweimal ein Hash-Wert des eingetippten Passwortes berechnet und erst dann mit dem in der Datenbank gespeicherten Hash-Wert verglichen.

Joomla!

- *Beschränkung von Dateiuploads*

Joomla! ermöglicht neben der Einstellung von erlaubten Dateitypen und -endungen, auch die Angabe der maximalen Uploadgröße einer Datei und auch die Angabe von explizit verbotenen Dateitypen.

Drupal

Drupal bietet sehr ausgereifte CMS-eigene Sicherheitsfeatures, welche im Folgenden skizziert werden.

- *Anzeige der Passwortstärke*

Bei der Installation wird die Stärke des verwendeten Administratorpasswortes je nach Länge und verwendeter Zeichen angezeigt.

- *Automatische Benachrichtigung bei wichtigen Updates per Email*
- *Automatische Filterung von benutzererstellten Inhalten*

Um zu vermeiden, dass von Benutzern der Seite Schadcode, etwa mit Hilfe von JavaScript, in das System eingeschleust wird, ermöglicht Drupal die Filterung jeglichen Inhalts. So kann zum Beispiel die Menge der verfügbaren HTML-Tags auf bestimmte, unschädliche eingeschränkt werden. JavaScript- und CSS-Tags werden automatisch gelöscht. PHP-Code ist ab Drupal 6 standardmäßig gar nicht mehr erlaubt, kann aber durch das Modul „PHP Filter“ aktiviert werden.

- *Beschränkung von Downloads*

Dateien können in Drupal direkt über den Webserver heruntergeladen werden. Es ist aber auch möglich, alle Downloads über Drupal zu leiten, sodass ein Download z.B. auf bestimmte Benutzergruppen bzw. Rollen limitiert bleibt.

- *Logs*

Drupal hat bereits im Zustand der Installation umfangreiche Logfunktionen integriert und aktiviert.

Beinahe jede Aktivität wird geloggt, seien es erfolgreiche oder fehlgeschlagene Login-Vorgänge, das Anlegen eines neuen Benutzers, Zugriffsversuche auf nicht vorhandene

oder geschützte Seiten, Änderungen der Konfiguration, das Öffnen und Schließen von Sessions, Inhaltsänderungen, etc...

Jeder Logeintrag wird gespeichert mit dem ausführenden Benutzer, Datum und Uhrzeit, Referer und IP-Adresse.

Fazit

Typo3 und Drupal bieten eine Reihe von eigenen Mechanismen an, um die Sicherheit des eigenen Produktes noch zu verbessern. Diese Mechanismen sind zwar bei beiden Produkten unterschiedlich, jede Maßnahme macht jedoch speziell für das Produkt und den Einsatzzweck Sinn. Besonders hervorzuheben sind die umfangreichen Logfunktionen beider Produkte. Sie ermöglichen eine lückenlose Überwachung der betriebenen Webseite und helfen so, Angriffsversuche zu erkennen und abzuwehren oder zumindest im Nachhinein den Angreifer und den Angriffsvektor leichter identifizieren zu können.

Joomla! bietet bei den CMS eigenen Sicherheitsfeatures nicht wirklich neue Features an. Joomla! wird dadurch zwar nicht unsicherer, aber eben auch nicht sicherer. Einzig echter Nachteil sind die fehlenden Logfunktionen. Typo3 und Drupal sind auch hier klare Sieger gegenüber Joomla!.

3.3.3.4. Verbreitung des Systems, Referenzen

Typo3

Weltweit sind über 290.000 Installationen von Typo3 in 45 Sprachen im Einsatz. Typo3 wird von internationalen Konzernen wie EADS, T-Systems oder ThyssenKrupp, von Bildungseinrichtungen wie der City University of New York oder der Universität Stanford und auch von vielen nationalen und internationalen Organisationen, wie der UNESCO eingesetzt. [TYP-07/5]

Drupal

Auch Drupal ist ein häufig eingesetztes Content Management System. Laut Zahlen von Dries Buytaert laufen derzeit zwischen 20.000 und 40.000 Websites auf Basis von Drupal.[BUY-07]

Die Liste der Benutzer von Drupal ist lang. NGOs wie Greenpeace sind ebenso darunter, wie die asiatische Seite der Sony Playstation und Sony BMG. Neben MTV und der Zeitung „Die Welt“ setzt auch die Firma LindenLab mit ihrer Community-Anwendung „Second Life“ auf Drupal.[DRC-08]

Joomla!

Zu den Referenzen von Joomla! zählen unter anderen der Webauftritt der Fachzeitschrift PCPraxis sowie Porsche Brasilien, es bleibt jedoch festzuhalten, dass Joomla! vor allem bei sehr vielen kleineren, privaten Webseiten zum Einsatz kommt.

3.3.4. Funktionen

Typo3

- *Newssystem*
Ja, ist durch Extension „tt_news“ sehr einfach realisierbar.
- *Übernahme von RSS-Feeds fremder Seiten in das eigene News-System*
Ja, durch diverse Extensions ist die direkte Übernahme von RSS-Feeds in tt_news möglich.
- *Volltextsuche*
Ja, die Suche in PDFs und Office-Dokumenten ist mit einer Extension möglich.
- *Automatische Bildverarbeitung (Stichwort „Thumbnails“)*
Ja, Typo3 bietet sehr ausgereifte und umfangreiche Bildverarbeitungsfunktionen durch Verwendung der Bibliotheken „GraphicsMagicks“ bzw. „ImageMagicks“.
- *Möglichkeit zum Dateupload*
Ja.
- *Workflowsystem zur Unterstützung des Dateuploads*
Ja, ist durch eine eigene Extension zu realisieren.

Drupal

- *Newssystem*
Ja, ist durch Blogsystem realisierbar.
- *Übernahme von RSS-Feeds fremder Seiten in das eigene News-System*
Ja.
- *Volltextsuche*
Ja.
- *Automatische Bildverarbeitung (Stichwort „Thumbnails“)*
Ja, ist durch ein Modul möglich.
- *Möglichkeit zum Dateupload*
Ja, kann ebenfalls durch ein Modul integriert werden.

- *Workflowsystem zur Unterstützung des Dateiuploads*
Ja, kann über Trigger bzw. Actions realisiert werden.

Joomla!

- *Newssystem*
Ja, Funktionalität ist durch Artikelverwaltung integriert.
- *Übernahme von RSS-Feeds fremder Seiten in das eigene News-System*
Ja.
- *Volltextsuche*
Ja.
- *Automatische Bildverarbeitung (Stichwort „Thumbnails“)*
Ja, einfache Funktionen sind integriert.
- *Möglichkeit zum Dateiupload*
Ja.
- *Workflowsystem zur Unterstützung des Dateiuploads*
Ja, ist durch Anpassungen der bestehenden Funktionen möglich.

Fazit

Prinzipiell sind mit allen getesteten CMS die Anforderungen von *security.fim.uni-linz.ac.at* zu erfüllen. Da die Anforderungen für das Webportal zum Teil sehr spezifisch sind, müssten alle drei Content Management Systeme zu einem gewissen Grad angepasst und erweitert werden. Prinzipiell sind in diesem Punkt die Systeme als gleichwertig zu betrachten. Berücksichtigt man die Extension „kickstarter“, welche die Erstellung von Erweiterungen sehr vereinfacht, so hat Typo3 im Sinne der Erweiterbarkeit leichte Vorteile.

3.3.5. Mehrsprachigkeit

Typo3

Typo3 bietet zwei unterschiedliche Konzepte für mehrsprachige Webseiten an: Das „two-tree“- und das „one tree fits all languages“-Konzept.

Grundlage des „two-tree“-Konzeptes ist die Erstellung von zwei oder mehr logisch getrennten Webseiten in unterschiedlichen Seitenbäumen, die die gleiche Wurzel besitzen und damit gemeinsam administrierbar sind. Auch die Wiederverwendung von Templates, Grafiken, etc. ist damit gewährleistet. Dieses Konzept der Trennung in voneinander unabhängige Seiten ist

vor allem dann sinnvoll, wenn die Webseite in den verschiedenen Sprachen mit verschiedenen Inhalten versehen bzw. anders strukturiert werden soll.

Das „one tree fits all languages“-Konzept geht einen anderen Weg. Jede Webseite in Typo3 wird ohne Zutun des Benutzers in einer Standardsprache, der „default“-Sprache, erstellt. Welche Sprache die „default“-Sprache sein soll, ist konfigurierbar.

Sollen nun zusätzliche Sprachversionen hinzukommen, so ist es erforderlich, in der Wurzel des Seitenbaumes zusätzlich zur Default-Sprache eine „Website-Language“ zu erstellen. Ab dem Zeitpunkt, wo zwei oder mehr „Website-Languages“ im Seitenbaum vorhanden sind, kann für jedes Inhaltselement einer Seite eine Übersetzung angefertigt werden, wobei diese direkt an eine Seite gekoppelt ist. Dies hat einige große Vorteile: Es ist sehr einfach, etwa über einen Menüpunkt oder einen Selektor, zwischen den Sprachversionen einer Seite umzuschalten. Da die Seitentitel ebenfalls übersetzt werden können, sind auch alle Menüeinträge in der gerade ausgewählten Sprache verfügbar. Weiters wird, wenn eine angeforderte Seite in der gewünschten Sprache nicht verfügbar ist, wahlweise einfach die gleiche Seite, allerdings in der Standard-Sprache, oder aber diese Seite im Menü überhaupt nicht angezeigt.

Drupal

Seit Drupal 6.0 ist die Funktion der Mehrsprachigkeit bereits in der Grundinstallation enthalten. Um die Funktionalität zu aktivieren, muss eine Standard-Sprache definiert werden. Es können beliebig viele weitere Sprachen definiert werden. Inhalte können dann bei der Erstellung entweder der Standard-Sprache oder anderen Sprachen zugeordnet werden. Die Übersetzungen von Inhaltselementen sind selbst wieder eigene Nodes.

Die Auswahl der Sprache kann entweder über Sprachpräfixe im Seitenpfad, die Einstellungen eines registrierten Benutzers, die Spracheinstellungen des Browsers oder über das Auslesen des Domainnamens der Seite (z.B. mit Subdomain de.seite.com) beeinflusst werden. Die Verwendung von Drop-Down-Elementen zur Sprachauswahl ist ebenfalls möglich.

Joomla!

In Joomla!-Kern selbst ist Mehrsprachigkeit für Webseiten nicht vorgesehen. Durch die Installation der Komponente „Joomfish!“ kann jedoch auch Joomla! mehrsprachige Webseiten verwalten. Joomfish! bietet im Backend die Möglichkeit, sämtliche Inhalte und sämtliche Joomla!-eigene Komponenten in beliebig viele Sprachen zu übersetzen. Auch die Ausgaben der Komponenten von Drittanbietern können übersetzt werden, sofern für sie so

genannte „Extensionfiles“ existieren. Diese enthalten Informationen zur Struktur der Inhaltselemente dieser Komponente. Joomfish! bietet zur Auswahl der Sprache ein Modul an, welches in die Webseite eingebunden werden kann. [JOO-08]

Fazit

Typo3 bietet gleich zwei Konzepte zur Mehrsprachigkeit, wobei für *security.fim.uni-linz.ac.at* vor allem das „one tree fits all languages“-Konzept interessant ist. Durch die enge Koppelung der Übersetzungen an die Inhalte einer Seite wird vor allem die Übersetzungsarbeit sehr erleichtert. Auch die einfache Umschaltmöglichkeit zwischen den Sprachen spricht für dieses Konzept.

Im Gegensatz dazu scheint die Mehrsprachigkeit in Drupal noch nicht ganz ausgereift zu sein. Die Übersetzungen von Inhalten sind eher lose an die Originale gekoppelt, außerdem ist die Erstellung von Menüs in mehreren Sprachen nicht ganz einfach möglich.

Joomla! bietet gar kein Sprachkonzept an, die Komponente „Joomfish!“ macht dieses Manko allerdings beinahe zur Gänze wieder wett. Übersetzungen stellen hier kein Problem dar.

Im Sinne der Mehrsprachigkeit ist Typo3 als Favorit zu betrachten, gefolgt von Joomla!. Das Konzept von Drupal scheint aus den genannten Gründen für *security.fim.uni-linz.ac.at* eher weniger geeignet.

3.4. Auswahl eines CMS

Nach einem Kurzüberblick über die Vor- und Nachteile jedes getesteten Content Management Systems wird schließlich die Entscheidung für eines der Systeme getroffen und begründet.

3.4.1. Typo3 – Vor- und Nachteile

Typo3 ist objektiv das mächtigste aller getesteten Systeme. Der Schwierigkeitsgrad für Entwickler und die nötige Einarbeitungszeit in das System sind dafür auch relativ hoch anzusetzen. Typo3 richtet sich an professionelle Entwickler, die bereit sind, Zeit zu investieren. Dafür sind die gebotenen Möglichkeiten vielfältig und umfangreich. Die Sicherheit des Systems ist bei entsprechender Konfiguration zu einem hohen Grad gegeben. Das Rechtesystem ist als sehr feingranular zu bezeichnen – ein Mehr an Differenzierung ist für die meisten Anwendungen nicht notwendig.

Die Möglichkeit, mit TypoScript eine eigene Skriptsprache zu verwenden, hat sowohl positive wie auch negative Auswirkungen. Einerseits erschwert TypoScript als eigenes

Konzept die Einarbeitung noch zusätzlich. Andererseits eröffnet die Sprache Möglichkeiten der eleganten und schnellen Konfiguration, die bei einem anderen CMS nicht möglich sind.

Besonders hervorzuheben sind auch die Konzepte zur Mehrsprachigkeit. Das Konzept „one tree fits all languages“ ist trotz seiner konzeptuellen Komplexität einfach und intuitiv zu bedienen und ist daher für *security.fim.uni-linz.ac.at* zu favorisieren.

3.4.2. **Drupal – Vor- und Nachteile**

Drupal ist auf Web 2.0 Anwendungen ausgerichtet und hat dort auch seine Stärken. Integrierte Benutzerregistrierung, automatische Einbindung und Generierung von Feeds, Kommentarfunktionen, Forensoftware, das Konzept des „Buches“ – sämtliche Features sind eindeutig auf Community-Webseiten ausgerichtet.

Sicherheit wird in Drupal groß geschrieben. Viele kleine Features, wie die Anzeige der Passwortstärke oder auch die automatische Benachrichtigung bei Updates per Email sorgen für ein Mehr an Sicherheit. Das Rollenmodell, in dem die Module ihre zuweisbaren Rechte anbieten, ist als innovativ zu bezeichnen. Ein Nachteil ist allerdings, dass diese Rechte global über eine ganze Seite angewendet werden und z.B. nicht nur für eine spezielle Node konfigurierbar sind.

Die Sprachenunterstützung in Drupal ist zwar vorhanden, erscheint allerdings konzeptuell noch etwas unausgegoren.

3.4.3. **Joomla! – Vor- und Nachteile**

Joomla! ist ein einfach zu bedienendes Content Management System, das trotzdem viele Funktionalitäten anbietet. Darin liegen auch klar die Vorteile dieses Content Management Systems. Zielgruppe von Joomla! sind dementsprechend kleinere Firmen bzw. Privatpersonen, welche sich schnell und einfach eine gute, qualitativ hochwertige Internetpräsenz schaffen wollen.

Konzeptuelle Nachteile, wie ein sehr begrenztes Sicherheitsmodell, eine fixe zweistufige Kategorisierung von Inhalten sowie fehlende Mehrsprachigkeit in der Basisversion müssen dafür in Kauf genommen werden.

3.4.4. **Die Entscheidung**

Als Content Management System zur Implementierung des Portals für den Studiengang „Netzwerke und Sicherheit“ wird auf Grund der oben getesteten Kriterien und der oben angeführten Vor- und Nachteile der jeweiligen Systeme **Typo3** ausgewählt.

Zusätzlich zu den genannten Gründen kann angeführt werden, dass am Institut für Informationsverarbeitung und Mikroprozessortechnik „Know-How“ über Typo3 vorhanden ist, was den Nachteil der langen Einarbeitungszeit bis zu einem gewissen Grad kompensiert.

4. Typo3 – Konzepte

Nach der überblicksmäßigen Beschreibung der Konzepte im vorigen Kapitel dient dieses Kapitel der genaueren konzeptuellen Betrachtung von Typo3.

4.1. Die Ordnerstruktur einer Typo3-Installation

Das Installationsverzeichnis von Typo3 enthält folgende relevante Ordner:

- *fileadmin*
Dieser Ordner ist der Stammordner aller im Backend hochgeladenen Dateien. Dies können HTML-Templates, CSS-Dateien und Bilder, JavaScripts oder andere sein – alle Dateien, auf die auch der Besucher der Webseite Lesezugriff haben muss. Im Backend kann nur auf diesen Ordner der Typo3-Installation direkt zugegriffen werden.
- *t3lib*
Enthält Typo3-Bibliotheken für Datenbankzugriff, Dateifunktionen, Benutzerauthentifizierung, etc.
- *typo3*
Dieser Ordner enthält den Source-Code und alle relevanten Dateien des Backends sowie die Datei *index.php*, welche das Backend rendert. Der Zugriff auf das Backend erfolgt über `<Domain-Name>/typo3`.
- *typo3conf*
typo3conf enthält alle Einstellungen für eine konkrete Typo3-Installation. Auch alle Erweiterungen werden in diesen Ordner installiert.
- *typo3temp*
Hier werden alle temporären Dateien, wie der Cache, aber auch bearbeitete Bilder gespeichert.
- *uploads*
Dies ist der zentrale Upload-Ordner für die Webseite. Hier werden z.B. die über Formulare hochgeladenen Dateien gespeichert.

Die Datei *index.php* liegt im Stammverzeichnis von Typo3. Sie ist verantwortlich für die Generierung der eigentlichen Webseite. [SKÅ-07/2]

4.2. Administration von Typo3

Jede mit Typo3 erstellte Webseite hat ihren Ausgangspunkt im Backend. Diese Oberfläche ist dreigeteilt. Neben der Menüleiste am linken Rand befindet sich rechts der Inhaltsbereich. Oberhalb, am rechten Seitenrand, sind einige häufig benötigte Funktionen wie Logout oder das Löschen der Caches angeordnet.

Über das Menü können folgende Funktionen aufgerufen werden:

- *Web*

Wird ein Unterpunkt dieses Menüeintrags gewählt, so wird das Inhaltsfenster rechts nochmals geteilt. Im linken Bereich dieses Fensters werden alle Seiten der administrierten Webseite in Baumstruktur angezeigt, der Inhalt des rechten Teils ist abhängig vom gewählten Untermenüpunkt und der gewählten Seite.

- *Page*

Hier werden die einzelnen Container für Inhaltselemente einer Seite und die enthaltenen Inhalte angezeigt. Diese können verändert oder gelöscht, bzw. neue hinzugefügt werden.

- *View*

Es wird eine Vorschau auf die gewählte Seite angezeigt.

- *List*

Diese Funktion zeigt eine Auflistung des Inhalts einer Seite an. Dies können neben den eigentlichen Inhaltselementen auch Frontendbenutzer der Seite, Seitentemplates oder andere Daten sein.

- *Info*

Info zeigt grundlegende Informationen zur gewählten Seite wie Ersteller, Erstellungs- und Änderungsdatum an.

- *Access*

Dieser Menüpunkt erlaubt die Konfiguration der Zugriffsberechtigungen für einzelne Seiten. Dabei wird unterschieden zwischen Rechten des Besitzers („Owner“), Rechten der besitzenden Gruppe sowie Rechten von allen Backendbenutzern. Folgende Rechte können auch rekursiv für die Unterseiten jener Seite vergeben werden:

„Seite anzeigen“ – „Inhalt ändern“ – „Seite ändern“ (z.B. Seitentitel, Sichtbarkeit, Archivdatum) – „Seite löschen“ – „Neue Unterseite erstellen“

- *Functions*

Hinter „Functions“ verbergen sich Assistenten, die es ermöglichen, häufig benötigte Aufgaben schnell auszuführen. Dazu zählen die Erstellung von bis zu neun Seiten auf einmal und die Sortierung von Seiten.

- *Versioning*

Das Konzept der „Workspaces“ erlaubt die Bearbeitung und Änderung von Webseiten, ohne dass die Änderungen sofort im Frontend aufscheinen. Dies hat den Vorteil, dass Änderungen vor der Veröffentlichung auf Fehler kontrolliert werden können. Der Menüpunkt „Versioning“ erlaubt den Zugriff auf die verschiedenen Versionen einer Seite.

- *Template*

In diesem Unterpunkt können Seitentemplates erstellt und administriert werden.

- *File*

- *Filelist*

Dieser Menüpunkt erlaubt den Zugriff auf alle für die konkrete Webseite relevanten Dateien. Diese sind, wie oben erwähnt, im Ordner *fileadmin* gespeichert. Dateien können im Backend in diesen Ordner bzw. Unterordner hochgeladen werden.

- *User*

- *Task Center*

In Typo3 kann durch eine Erweiterung ein „Task Management System“ hinzugefügt werden. So lassen sich beispielsweise „Workflows“ zur Bearbeitung von Seiten oder zum Erstellen von Nachrichten einrichten. Im Auslieferungszustand bietet Typo3 jedoch lediglich eine Schnittstelle zu diesem System.

- *User settings*

Dies erlaubt die Konfiguration des eigenen User-Accounts, der Backend-Sprache sowie zahlreicher anderer Basiseinstellungen des eigenen Accounts. Administratoren können auch die Einstellungen anderer Benutzer ändern.

- *Workspace*

Wie bereits erwähnt, bietet Typo3 mächtige Versionierungsfunktionen an. Eine davon ist der so genannte „Draft-Workspace“, bei dem Änderungen, im Gegensatz zum „Live-Workspace“, nicht in das Frontend durchgeschaltet werden. So kann zum Beispiel ein komplettes Neudesign einer Seite im Draft-Workspace durchgeführt werden, ohne den laufenden Betrieb zu stören, oder aber auch neue Funktionalität vor der Freischaltung ausgiebig getestet werden. Mit einem Knopfdruck können alle Änderungen in den Live-Workspace und damit in den laufenden Betrieb übernommen werden.

Seit der Version 4.2 können beliebig viele Workspaces verwaltet werden.
- *Admin tools*
 - *User Admin*

Dieses Tool gibt eine Übersicht über alle vorhandenen Backend-Benutzer und erlaubt die Anzeige von Statistikdaten und auch die Konfiguration der Benutzer. Typo3 ermöglicht es Administratoren, mit einem Klick andere Backendbenutzer zu simulieren. Damit kann leicht festgestellt werden, ob die getätigten Einstellungen auch die gewünschten Effekte hervorrufen.
 - *ExtManager*

Der „Extension Manager“ ist eines der wichtigsten Werkzeuge in Typo3. Er ermöglicht den Download, die Installation und die Wartung der Erweiterungen von Typo3.
 - *DB check*

Dieser Menüpunkt erlaubt diverse Überprüfungen der internen Datenbankrelationen und enthält auch eine Suchfunktion.
 - *Configuration*

Erlaubt die Ansicht, jedoch nicht die Änderung der beiden wichtigen Konfigurationsdateien „\$TCA-Array“ und „\$TYPO_CONF_VARS“. Wichtig ist dieses Feature vor allem bei der Erstellung einer Seite, da man an einer zentralen Stelle sämtliche Variablenbelegungen abfragen kann.
 - *InstallTool*

Das InstallTool erlaubt die grundlegende Einrichtung von Typo3. Datenbankbindung, Konfiguration des Bildverarbeitungssystems („ImageMagick“ bzw. „GraphicsMagicks“) und vieles mehr.

Das InstallTool kann nach erfolgreicher Einrichtung von Typo3 deaktiviert werden, um Missbrauch vorzubeugen.

- *Log*

Typo3 besitzt umfangreiche Logging-Funktionalität. Neben Logins bzw. Loginversuchen werden auch Änderungen an Einstellungen und Inhalten gespeichert. Änderungen an der Datenbank, sprich, auch inhaltliche Änderungen, können jederzeit rückgängig gemacht werden.

- *Indexing*

„Indexing“ ermöglicht die Anzeige des Indizierungsstatus der Seite. Diese Funktion ist besonders für die Suche im Frontend von Bedeutung.

- *Help*

Die Punkte „About modules“, „About“ und „Manual“ unterhalb von Help enthalten Informationen und Hilfe.

4.3. TypoScript – die Typo3-eigene Skriptsprache

„TypoScript“, kurz „TS“, ist eine Skriptsprache, mit der Informationen in einfacher Art und Weise abgespeichert werden können. Jeder TypoScript-Code wird zur Laufzeit in ein multidimensionales PHP-Array umgewandelt. Theoretisch könnte also die Konfiguration von Typo3 direkt in PHP erfolgen, TS vereinfacht die Konfiguration durch die einfachere Syntax jedoch wesentlich. [SKÅ-07]

4.3.1. Syntax

TypoScript dient in erster Linie dazu, Typo3 zu konfigurieren. Meist werden also Werte an Variablen zugewiesen. Dies geschieht in der Form

```
variable = wert
```

TypoScript unterscheidet zwischen Objekten und ihren Eigenschaften, wobei die Übergänge oft fließend sind. Ein Beispiel verdeutlicht dies:

```
objekt.hintergrund = blau
objekt.hintergrund.transparenz = 95%
```

Hier ist `hintergrund` sowohl Eigenschaft als auch Objekt.

Die Interpretation von TypoScript erfolgt zeilenweise.

4.3.2. Operatoren

TypoScript bietet folgende Operatoren an: =, <, >, {, (, <Leerzeichen>

- *Zuweisung* „=“

Dieser Operator weist der Eigenschaft bzw. dem Objekt links den Wert rechts zu. Alles rechts von „=“ bis zum Ende der Zeile wird als Wert betrachtet.

- *Änderung durch Funktion* „:=“

Mit Hilfe dieses Operators kann ein vorher zugewiesener Wert durch eine vordefinierte Funktion geändert werden. Rechts von „:=“ steht die Änderungsfunktion mit eventuellen Argumenten. Typo3 bietet einige vordefinierte Funktionen, es besteht aber auch die Möglichkeit, eigene Funktionen zu definieren.

Beispiel:

```
myObject = TEXT
myObject.value = 1,2,3
myObject.value := addToList(4,5)
myObject.value := removeFromList(2,1)
```

- *Schachtelung durch* „{..}“

Der Einsatz von geschwungenen Klammern erlaubt die Schachtelung des Codes. Die Übersichtlichkeit wird dadurch erhöht und der Schreibaufwand minimiert.

Beispiel:

```
objekt.hintergrund = blau
objekt.hintergrund.transparenz = 95%
objekt.hintergrund.eigenschaft = wert
```

bedeutet das gleiche wie

```
objekt{
    hintergrund = blau
    hintergrund{
        transparenz = 95%
        eigenschaft = wert
    }
}
```

- *Mehrzeilige Werte*: „(...)“

Mit Hilfe von runden Klammern können Variablen mehrzeilige Werte zugewiesen werden. Im folgenden Beispiel wird ein mehrzeiliger HTML-Code zugewiesen.

```
myObject = HTML
myObject.value (
<BLINK>
HTML - code
</BLINK>
)
```

- *Kopieren* „<“

Dieser Operator kopiert ein gesamtes Objekt, inklusive aller Eigenschaften und Werte in ein anderes Objekt. Dieses wird komplett überschrieben.

Anzumerken ist, dass der Operator „Deep Copies“ anfertigt. Es existieren nach der Kopie wirklich **zwei** gleiche Objekte, nicht nur eine Referenz.

- *Löschen* „>“

Durch diesen Operator wird ein Objekt komplett gelöscht.

- *Bedingungsoperator* „[...]“

Als spezielles Feature enthält TypoScript eine Art „if-Ersatz“, nämlich „Bedingungen“. Zeilen zwischen der Bedingung sowie der Zeile [GLOBAL] werden nur im Falle der erfüllten Bedingung ausgeführt. Werden in einer Zeile mehrere Bedingungen angegeben, so werden diese mit dem „or“-Operator verknüpft. Bedingungen müssen außerhalb von geschwungenen Klammern stehen. Es existiert auch eine [ELSE]-Anweisung, welche durch [END] bzw. [GLOBAL] abgeschlossen werden kann.

Die Bedingung im folgenden Beispiel prüft, ob der verwendete Browser „Netscape“ ist und gibt dementsprechend einen Text aus. In der letzten Zeile des Beispiels wird der Text umschlossen von | fungiert in TypoScript als Platzhalter.

```
page.typeNum=0
page = PAGE
page.10 = TEXT
[browser=netscape]
    page.10.value = Netscape
[else]
    page.10.value = Not a netscape browser!
[end]
page.10.wrap = <b>|</b>
```

Wie im Beispiel klar ersichtlich ist, besitzt TypoScript gewisse vorreservierte Wörter. Neben „browser“ gibt es noch viele andere Möglichkeiten, Konditionen vorzunehmen.

Aus dem angeführten Beispiel kann man ebenso erkennen, dass TypoScript kontextsensitiv ist. „page“ ist eine Variable, PAGE die aktuell generierte Seite. Grundsätzlich gilt: großgeschriebene Wörter sind von Typo3 reserviert. So gibt es HTML, TEXT, IMAGE, etc...

4.3.3. Sprachelemente in TypoScript

TypoScript bietet verschiedene Arten von Sprachelementen an. Die folgenden Absätze beschreiben die wichtigsten:

- *Bedingungen*

Wie im vorigen Punkt behandelt, ist in TypoScript eine Art „if-Abfrage“ an Hand von Bedingungen möglich. Vergleichswerte sind neben „browser“, „system“, „IP“ auch zeitliche Abfragen wie „month“, „hour“ und Berechtigungsprüfungen wie „usergroup“.

Externe Funktionen lassen sich ebenfalls zu Überprüfungen von Werten heranziehen.

- *Funktionen*

In TypoScript gibt es eine Reihe von vordefinierten Funktionen. Beispiele sind „textStyle“, welche Texte formatiert, „filelink“, welche Dateifunktionen anbietet und „parseFunc“, mit der Inhalte geparkt werden können.

- *cObjects*

Mit Hilfe der „cObjects“ (contentObjects) können verschiedene Elemente einer Seite direkt erzeugt bzw. angesprochen werden. cObjects werden groß geschrieben.

Beispiele sind HTML, TEXT, IMAGE, FILE und auch HMENU, welches ein hierarchisches Menü erzeugt. Jedes cObject bietet verschiedene Eigenschaften an, so bietet TEXT unter anderen die Eigenschaft „value“.

Ein neues cObject kann einfach durch die Zuweisung des entsprechenden Typs an eine Variable angelegt werden.

```
page.10 = TEXT
page.10.value = anzuzeigender Text
```

- *Gifbuilder*

Typo3 bietet mächtige Funktionen zur Grafikbearbeitung bzw. auch zur Erstellung von gif-Bildern aus Text. Diese Funktionen sind im Gifbuilder zusammengefasst. Funktionen wie Skalierung, Zuschneiden, Schatten oder diverse Effekte können auf Bilder sowie auf in Bilder umgewandelte Texte angewendet werden. So lassen sich beispielsweise Seitenüberschriften grafisch gestalten oder auch Menüs komplett als Grafik erzeugen.

- *MENU Objects*

HMENU als cObject enthält immer ein MENU-Objekt, mit dem die Art des Menüs festgelegt wird. Mögliche Werte sind TMENU, welches ein textuelles Menü generiert,

oder auch GMENU, welches ein grafisches Menü mit dem Gifbuilder generiert. Genauere Informationen zur Menüerstellung in TypoScript sind in Abschnitt 4.3.5 zu finden.

4.3.4. Einsatz von TypoScript

Folgende Zeilen geben einen Überblick über die Einsatzgebiete von TypoScript in Typo3.

4.3.4.1. Templates

TypoScript findet hauptsächlich in den so genannten „Template Records“ Verwendung. In Typo3 benötigt jede Seite einen Template Record, welcher einerseits angibt, wie die im Seiteneditor erstellten Inhalte dieser Seite angezeigt werden sollen und es andererseits erlaubt, die Konfiguration der Seite bzw. der in der Seite enthaltenen Plugins zu steuern. Da es heute moderne Ansätze gibt, die die Zuordnung von Inhalten zu einer Seitenvorlage wesentlich vereinfachen (siehe Abschnitt 4.5, TemplaVoila), ist der erste oben angesprochene Zweck von TypoScript mehr oder weniger obsolet. Die Templates sind seitdem sehr viel weniger komplex. Die Konfiguration einer Seite bzw. der enthaltenen Erweiterungen muss jedoch nach wie vor zum Großteil mittels TypoScript geschehen.

Ein Template besteht aus mehreren Teilen:

- *Constants*
Hier können Konstanten definiert werden, welche später im Template auch mehrfach verwendet und eingefügt werden können.
- *Setup*
Enthält den eigentlichen Code nach der oben vorgestellten TypoScript-Syntax.
- *Include static*
Ursprünglich war direkt in einem Template die Information enthalten, wie die Inhalte einer Seite strukturiert und angezeigt werden sollen. Mit Hilfe der „Include static“-Einstellung können vorgefertigte Vorlagen einfach eingebunden werden.
Heute kann auf diese Art und Weise ein Standard-Stylesheet für die Ausgabe von Typo3-eigenen Elementen eingebunden werden.
- *Resources*
In einem Template können verschiedenste Elemente, wie Bilder oder Stylesheets verwendet und angesprochen werden. Mit Hilfe der „Resources“-Einstellung können diese Elemente bereits im Vorfeld definiert werden. Dies sorgt für eine zusätzliche Abstraktionsebene zum Dateisystem. Sämtliche Resources werden in einen Typo3-

eigenen Ordner kopiert und von dort abgerufen, was den Vorteil bietet, dass bei irrtümlicher Löschung oder Bearbeitung des Originals die Ausgabe nicht zerstört wird.

- *Weitere Einstellungen*

Wie für jedes Element können auch für Templates Start- und Enddatum der Gültigkeit definiert werden.

Templates von Unterseiten erben automatisch die Einstellungen der Templates von Seiten niedrigerer Tiefe. Es kann jedoch in einem Template explizit festgelegt werden, dass „Setup“ und „Constants“ von anderen Templates nicht verwendet werden sollen, ebenso wie das Template der Root-Seite einer Webseite als „Root-Template“ deklariert werden muss.

Die Vererbung von Templates hat den Vorteil, dass man pro Webseite in der Theorie nur ein Template benötigt, denn wenn die Rootseite ein Template besitzt, benutzt jede Unterseite automatisch auch dieses Template, was für alle seitenweit gleichen Einstellungen und Inhalte, wie Menüs, Suchboxen, Kopf- und Fußzeilen, etc. sehr praktisch ist.

Zur besseren Übersichtlichkeit können Templates auch geschachtelt werden, bzw. können in ein Template beliebig viele weitere Templates eingebunden werden. Das erlaubt die Auslagerung von Templates in einen einen SysFolder. Die Rootseite muss allerdings trotzdem ein Template besitzen, in welches die anderen Templates eingebunden sind.

Einstellungen in einem Template gelten immer so lange, bis sie von einem Template überschrieben werden, welches sich „näher“ an der gerade anzuzeigenden Seite im Sinne der Tiefe des Seitenbaumes befindet.

Theoretisch können auch Inhalte wie Grafiken, Texte, Tabellen, etc. direkt in Templates mit Hilfe von TypoScript erstellt werden. Allerdings werden wohl die meisten Redakteure und auch Entwickler die Möglichkeiten des Seiteneditors mit dem integrierten RichTextEditor und den vielfältigen Konfigurationsmöglichkeiten bevorzugen.

4.3.4.2. PAGE-TSconfig

Zusätzlich zu den oben genannten Templates besteht die Möglichkeit, direkt in den Seiteneigenschaften einer Seite grundlegende Einstellungen mit Hilfe von TypoScript vorzunehmen. Der Unterschied zu den Templates besteht im Wesentlichen darin, dass sich die PAGE-TSconfig Einstellungen auf Backendfelder beziehen, während Templates vor allem Einfluss auf die Ausgabe im Frontend nehmen.

PAGE-TSconfig ermöglicht z.B. die Konfiguration des Typo3-eigenen RichText-Editors. Auch manche Backend-Extensions, wie „crawler“, müssen über die PAGE-TSconfig

konfiguriert werden. TypoScript Konfiguration in der PAGE-TSconfig wird automatisch auf alle Unterseiten vererbt, Einstellungen in der Rootseite eines Webauftritts gelten also automatisch auch für alle anderen Seiten.

4.3.4.3. USER-TSconfig

Jeder Backend-Benutzer und jede Backend-Benutzergruppe kann mit Hilfe von TypoScript konfiguriert werden. So kann z.B. der RichText-Editor für bestimmte Benutzer deaktiviert oder auch die Option zum Löschen des Caches ein- bzw. ausgeblendet werden. Auch die Möglichkeit der Bearbeitung von Inhalten im Frontend wird hier konfiguriert.

Die tatsächliche TS-Konfiguration eines Benutzers ergibt sich aus den kumulierten Einstellungen aller Gruppen und des Benutzers selbst, wobei Einstellungen direkt beim Benutzer die Gruppeneinstellungen überschreiben.

4.3.5. Menüerstellung in TypoScript

Menüs können in TypoScript komplett dynamisch erzeugt werden. Die Menüerstellung erfolgt normalerweise auch im Root-Template der Webseite, da ein Menü üblicherweise auf der ganzen Webseite gleich sein soll.

Prinzipiell erfolgt die Erstellung eines hierarchischen, also mehrstufigen Menüs durch die Zuweisung des cObjects HMENU an den Marker, der das Menü enthalten soll, also entweder durch

```
marks.markername = HMENU
```

bzw. bei Verwendung von TemplaVoila

```
lib.markername = HMENU
```

Beginnend bei `markername.0` können die Ebenen im Seitenbaum angesprochen werden, um Menüs zu erzeugen. Durch die Eigenschaft `entryLevel` von HMENU kann bestimmt werden, welche Ebene im Seitenbaum `markername.0` entspricht.

Welche Art von Menü nun durch TypoScript generiert wird, hängt davon ab, welches „menuObject“ einem Menülevel zugeordnet wird.

- GMENU erzeugt ein grafisches Menü, bestehend aus einzelnen GIF-Bildern.
- GMENUFOLDOUT erzeugt ein mehrstufiges, grafisches Menü, bei dem die höheren Stufen dynamisch ausgeklappt werden können.
- TMENU erzeugt ein textuelles Menü.
- IMAGEMENU erzeugt als Menü ein großes GIF-Bild, wo die einzelnen Einträge durch Imagemaps definiert sind.

- JSMENU erzeugt ein Dropdownmenü, das mit JavaScript funktioniert.

Aus Gründen der Maschinenlesbarkeit sowie der Barrierefreiheit einer Webseite empfiehlt es sich, vorwiegend TMENUS zu verwenden. Weitere Informationen zur Menüerstellung sowie zu den konkreten Menüs auf der Security-Plattform sind in Abschnitt 6.5 zu finden.

4.4. Seitenerstellung

Die Gliederung einer Webseite wird, wie weiter oben erwähnt, in Typo3 mit Hilfe des „Seitenbaums“ festgelegt. Dieser ähnelt sehr stark dem Konzept der Ordner in Windows.

Die Wurzel, der sogenannte „Root-Knoten“, erkennbar in Abbildung 4 am Weltkugel-Icon, ist keine Seite im eigentlichen Sinne. Er enthält neben allen anderen Seiten noch globale Objekte wie Backend-Benutzer und sämtliche Sprachen.

Der Root-Knoten ist nicht nur die Wurzel aller im Frontend sichtbaren Webseiten, er enthält alle in der Datenbank gespeicherten Informationen in so genannten Sysfoldern, in der Abbildung rechts als graue Ordnersymbole erkennbar. Sysfolder sind Container für Elemente wie Frontend-Benutzer, Newseinträge, Templates oder auch Seiten. Jeder Eintrag im Seitenbaum hat eine eindeutige ID, durch die z.B. die Seiten im Frontend eindeutig angesprochen werden können. Seiten werden in Typo3 mit der Syntax `<URL>/index.php?id=<page_id>` geladen.

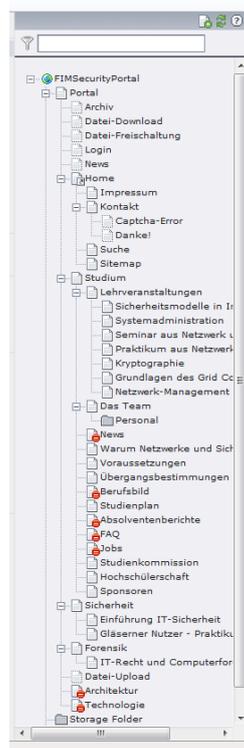


Abbildung 4: Seitenbaum in Typo3

Um eine Seite im Typo3-Backend neu zu erstellen, klickt man auf das Icon links neben einer Seite. Dabei wird ein kontext-sensitives Menü aufgerufen. Hier wird durch die Auswahl von „New“ eine neue Seite als Unterseite angelegt. Anschließend können die Seiteneigenschaften mit Hilfe einer graphischen Maske konfiguriert werden. Seit der Version 4.2 sind die Einstellungen in verschiedene „Reiter“ gegliedert, dies erhöht die Übersichtlichkeit.

Typo3 bietet für verschiedene Bedürfnisse verschiedene Arten von Seiten an. Je nach Auswahl ändern sich auch die Konfigurationsmöglichkeiten der Seite. Typo3 bietet folgende Typen an:

- *Standard*
Normale Frontendseite mit Inhalten, der am meisten verwendete Seitentyp.
- *Backend User Section*
Wie Standard, allerdings wird die Seite nur eingeloggten Backendbenutzern angezeigt.
- *Shortcut*
Mit diesem Typ kann in der Menüstruktur ein Link auf eine andere Seite eingefügt werden. Bei Aufruf wird die verlinkte Seite angezeigt.

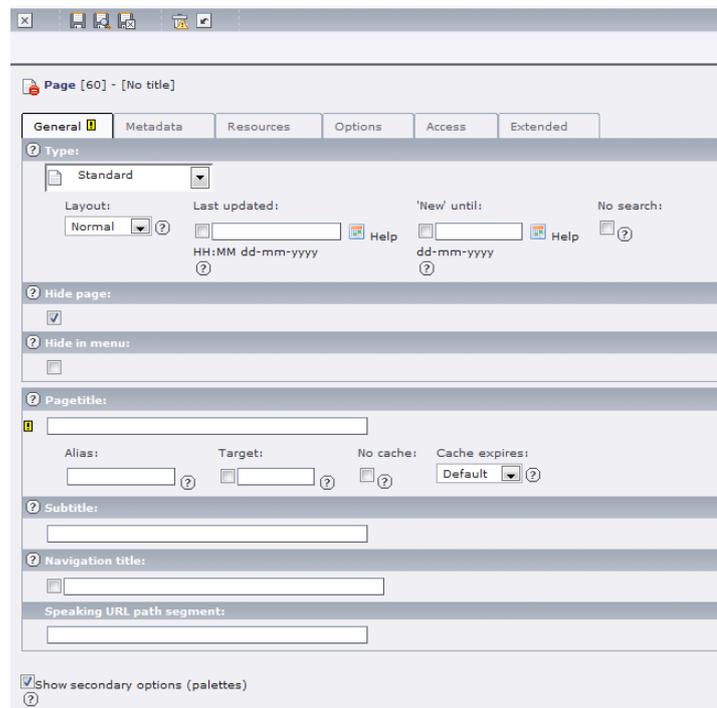


Abbildung 5: Konfiguration einer neuen Seite

- *Mount Point*
Effekt wie Shortcut, allerdings verweist der Menüeintrag auf eine eigene Seite, in der nur die Inhalte des Mount Points übernommen werden. Der Unterschied ist z.B. für

die „Breadcrumb“ von Bedeutung. Außerdem werden alle Unterseiten der gemounteten Seite in die Menüstruktur übernommen.

- *Link to external url*

Mit diesem Seitentyp kann eine externe URL in den Inhaltsbereich der eigenen Seite eingebunden werden.

- *SysFolder*

Wie weiter oben beschrieben, dienen SysFolder als Container für verschiedene Seitenobjekte.

- *Visual menu seperator*

Dieser Typ ist nur für die Menüerstellung relevant. Hier kann in der Menüstruktur eine optische Trennung eingefügt werden.

Die Seitentypen können durch Extensions ergänzt bzw. verändert werden.

Weitere Einstellungsmöglichkeiten einer Seite sind:

- *Hide page*

Diese Eigenschaft verhindert, dass die Seite im Frontend angezeigt wird. Weder im Menü noch bei direkter Eingabe der Seiten-ID wird die Seite angezeigt. Bei ihrer Erstellung wird jede Seite automatisch versteckt, da die Inhalte erst erstellt werden müssen.

- *Hide in menu*

Sorgt dafür, dass die Seite nicht im Menü angezeigt wird, aber trotzdem im Frontend verfügbar ist. Denkbare Anwendung ist hier eine Folgeseite bei Kontaktaufnahme über das Kontaktformular.

- *Pagetitle*

Der Seitentitel ist sowohl der interne Name einer Seite als auch der Titel im Head der generierten HTML-Seite.

- *Subtitle*

Wenn benötigt, kann hier ein Seitenuntertitel konfiguriert werden.

- *Navigation title*

Im Falle, dass sich der Menüeintrag einer Seite vom Seitentitel unterscheiden soll, kann hier der alternative Menüeintrag definiert werden.

Zahlreiche Metadaten, wie Autorname und Email-Adresse, Kurzfassung, Schlüsselwörter und Beschreibung lassen sich definieren. Ressourcen, wie Bilder oder andere Dateien lassen sich direkt zur Seite hinzufügen. Diese werden in einen eigenen Ordner kopiert und bleiben so

auch bei unabsichtlicher Löschung bestehen. Jede Seite kann zusätzlich mit TypoScript in der „PAGE-TSconfig“ konfiguriert werden.

Eine weitere wichtige Einstellung ist die „General Records Storage Page“. Das ist jene Seite, in welcher die grundlegenden, seitenweiten Einstellungen und Objekte gespeichert sind, wie Templates oder Frontend-Benutzer. Auch diese Einstellung wird jedoch vererbt, sodass im Prinzip nur die Wurzelseite diese Einstellung eindeutig festgelegt haben muss.

Auch grundlegende Spracheinstellungen können bereits bei der Seiteneinstellung getätigt werden. So kann beispielsweise eine Seite so konfiguriert werden, dass sie im Menü versteckt wird, wenn keine Übersetzung in der aktuell gewählten Sprache verfügbar ist.

Im Reiter „Access“ können grundlegende Einstellungen zu den Zugriffsrechten getätigt werden, so können Seiten bestimmten Benutzergruppen zugewiesen werden und einiges mehr.

Außerdem kann hier Datum und Zeit der Veröffentlichung bzw. des Offline-Stellens der Seite angegeben werden.

Der Reiter „Extended“ wird von TemplaVoila hinzugefügt. Dort können Einstellungen zu den verwendeten Templates getätigt werden. Dies wird im folgenden Kapitel 4.5 genauer erläutert.

4.5. Templating mit TemplaVoila

Die Struktur ist neben der inhaltlichen Gestaltung und dem Design die dritte große Säule einer Webseite. Die Struktur einer Seite wird in modernen Typo3-Systemen mit Hilfe von TemplaVoila erstellt. TemplaVoila macht im Grunde nichts anderes als eine fixe Zuordnung, ein „Mapping“, zwischen den „HTML-Tags“ einer HTML-Vorlage und von Typo3 erzeugten Inhalten.

Dieses praktische Werkzeug, das allerdings noch mehr Funktionen bietet, wird in den folgenden Absätzen behandelt.

4.5.1. Struktur eines Templates

Jedes Template ist in dem vorgegebenen Format <T3DataStructure> in XML codiert. Da jedes Template in einer Seite oder einem SysFolder der Seitenstruktur einer Webseite gespeichert sein muss, ist es auch automatisch in der Datenbank gespeichert.

Ein Template besteht aus den hierarchisch organisierten Teilen Data Structure (DS) und Template Object (TO).

In der Data Structure ist die Information enthalten, welche Felder mit welchen Eigenschaften man zu einem HTML-Template mappen kann, sprich, welche Felder man mit Inhalten füllen können soll. Im Template Object wird die Information gespeichert, welches Feld einer DS wohin und auf welche Art und Weise in eine konkrete HTML-Vorlage gemappt werden soll. Eine DS kann daher mehrere TOs enthalten, weil ein- und dasselbe Feld ja in verschiedenen aufgebaute HTML-Vorlagen eingebettet werden kann.

Prinzipiell ist es natürlich möglich, ein komplettes Template mit einem reinen Texteditor zu erstellen, weil alles in XML abgespeichert wird. Allerdings ist gerade die graphische Benutzeroberfläche eine der großen Stärken von TemplaVoila, weil sie die Erstellung von Templates sehr vereinfacht.

4.5.2. Arten von Templates

TemplaVoila ermöglicht die Erstellung von zwei Arten von Templates. Einerseits das bereits bekannte Seitentemplate, „page template“, also eine Vorlage und Struktur für eine Seite.

Andererseits bietet TemplaVoila auch die Möglichkeit, HTML-Vorlagen als „Content Element“ zu speichern. Diese können später als Inhaltselement in eine Seite eingefügt werden. Sinnvoll ist der Einsatz von Vorlagen als Content Element bei der Erstellung von gleich strukturierten Elementen mit unterschiedlichem Inhalt, z.B. könnte in einem Onlineshop die Struktur der Präsentation eines Artikels mit Hilfe von TemplaVoila realisiert werden.

4.5.3. Die Erstellung eines Templates

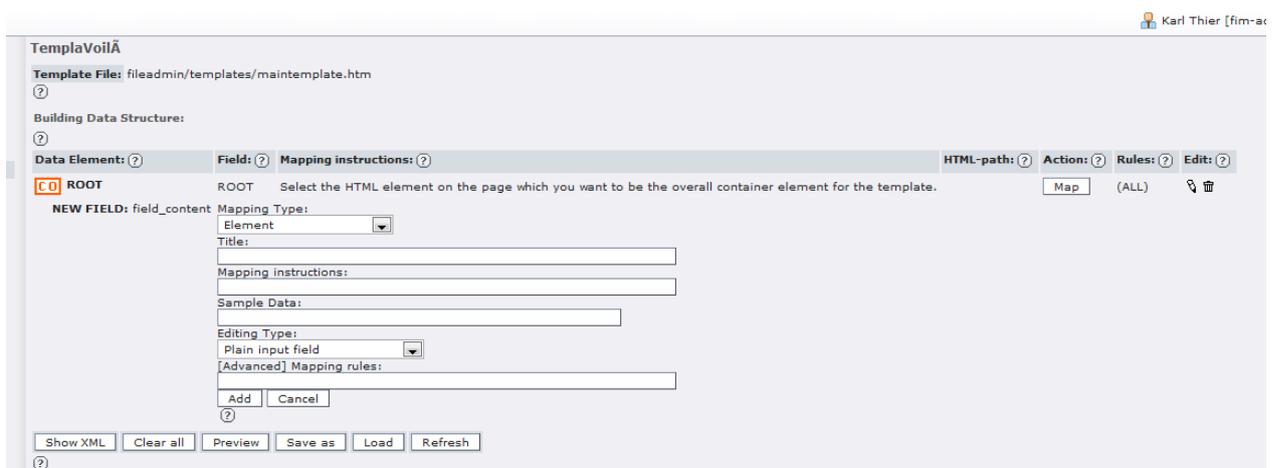


Abbildung 6: Erstellung der Data Structure in TemplaVoila

Die gewünschte Templatevorlage muss im Backend von Typo3 lesbar sein – muss also innerhalb des Ordners */fileadmin/* abgespeichert sein.

Durch die Installation von TemplaVoila wird im kontextsensitiven Menü ein Menüpunkt „TemplaVoila“ hinzugefügt. Wird nun auf eine Datei geklickt, welche von TemplaVoila verarbeitet werden kann, wie eben **.htm*, **.html*, etc., so wird dieser Menüpunkt angezeigt. Wird diese Option schließlich ausgewählt, startet TemplaVoila den graphischen Editor.

Bevor mit dem Mapping begonnen werden kann, muss erst die Data Structure definiert werden, sprich, welche Felder bzw. Feldtypen überhaupt gemappt werden sollen. Jede DS besitzt ein „Root-Element“, welches alle anderen Felder enthält. Deren Namen sollen nach der Konvention immer mit *field_* beginnen.

Wie in Abbildung 6 erkennbar, kann ein Feld verschieden konfiguriert werden.

Folgende Optionen stehen zur Verfügung:

- *Mapping Type*

Der Mapping Type gibt an, wie und was das Element später im HTML-Template mappen soll. Änderungen des Mapping Type haben direkte Auswirkungen auf die anderen Felder. Es gibt drei verschiedene Mapping Types:

- *Element*

Der Inhalt des HTML-Tags wird komplett ersetzt – Standard.

- *Container for Elements*

Durch die Auswahl dieses Mapping Types wird festgelegt, dass innerhalb dieses HTML-Tags weitere Tags gemappt werden – Root ist solch ein Container.

Diese Einstellung löscht alle Optionsfelder außer „Title“ und „Mapping Instructions“. Zusätzlich kann ein „Container for Elements“ als „Section“ markiert sein. Dies bedeutet, dass sich die darin befindlichen Inhaltselemente wiederholen können.

- *Attribute*

Mit diesem Mapping Type wird nicht der Inhalt eines HTML-Tags der Vorlage sondern nur ein Attribut desselben ersetzt. Welches dies konkret ist, kann erst beim Mapping ausgewählt werden.

- *Title*

Titel des Feldes.

- *Mapping instructions*

Anweisungen für jenen Bearbeiter, der die Zuordnung der Felder zur konkreten HTML-Vorlage durchführt.

- *Sample Data*
Hier können Beispieldaten eingetragen werden.
- *Editing Type*
Der Editing Type eines Feldes bestimmt, wie später dynamische Inhalte in das Feld eingefügt werden sollen bzw. welchen Typs diese sind.
Es gibt folgende Typen:
 - *Plain input field*
Einzeiliges Textfeld.
 - *Header field*
Hier wird eine Überschrift ausgegeben.
 - *Header field, graphical*
Prinzipiell gleicher Inhalt wie „Header field“, allerdings wird eine Graphik eingebunden.
 - *Text area for bodytext*
Wie das Textfeld, allerdings mehrzeilig. Keine Formatierung erlaubt.
 - *Rich text editor for bodytext*
Inhaltseingabe mit Hilfe des Rich Text Editors, Text vollständig formatierbar.
 - *Link field*
Hier kann ein Linktext sowie ein Link eingegeben werden.
 - *Integer value*
Ganze Zahl.
 - *Image field*
Bild bzw. Graphik.
 - *Image field, fixed W+H*
Bild bzw. Graphik mit fixen Abmessungen.
 - *Content elements*
Hier kann jedes im Backend verfügbare Inhaltselement eingefügt werden, wie Texte, Bilder oder Plugins. Sämtliche vorher vorgestellten Editing Types können mit diesem Editing Type auch erstellt werden.
 - *Selector box*
Erlaubt aus einer Reihe von vordefinierten Optionen die Auswahl einer Option.
 - *NONE*
Es wird nichts angezeigt und daher auch in der Vorlage nichts eingefügt.

- *TypoScript Object Path*

Inhalte können hier mit Hilfe von TypoScript eingefügt werden. Sinnvoll ist dies z.B. für die Menügestaltung oder auch die Einbindung von Plugins. Diese Einstellung erfordert zusätzlich die Eingabe eines Pfadnamens, mit dem später im Template mittels `lib.<pfadname>` die Stelle angesprochen werden kann.

Alle Editing Types sind im Seiteneditor bearbeitbar, außer dem „TypoScript Object Path“, der in einem Template mit Inhalten gefüllt wird.

- *[Advanced] Mapping rules*

Hier kann vorgegeben werden, wo in Bezug auf den gewählten HTML-Tag später das Mapping erfolgen soll. Dies kann innerhalb des gewählten Tags (Einstellung *:inner) erfolgen, wobei der Tag unverändert aus der Vorlage übernommen wird. Das Mapping kann aber auch außerhalb des Tags erfolgen (Einstellung *:outer), was bedeutet, dass der Tag aus der Vorlage gelöscht und vollständig durch den gewünschten Inhalt ersetzt wird.

In derselben Art und Weise werden alle zu mappenden Felder bzw. die dem Template zugrunde liegende Data Structure erstellt.

Anschließend kann durch Klick auf die Schaltfläche „Map“ rechts neben jedem erstellten Feld mit der Zuordnung des Feldes zu einer konkreten HTML-Vorlage begonnen werden.

Felder höherer Ordnung müssen natürlich zuerst gemappt werden, begonnen wird daher immer mit dem Root.



Abbildung 7: Mapping eines Feldes in TemplaVoila

Wie in Abbildung 7 zu sehen, erkennt TemplaVoila automatisch alle passenden HTML-Tags und bietet sie zum Mapping an. „Root“ wird meist auf den „body-Tag“ gemappt, was durch einen einfachen Klick auf den oben braun gefärbten Tag erfolgt. In einem zweiten Schritt wird die Auswahl der Mapping-Art getroffen, also ob außerhalb oder innerhalb des Tags gemappt werden soll. Alternativ kann auch ein „Range“ gemappt werden. Ein Range ist definiert als eine Folge von Tags auf der gleichen Ebene. Ist bei der Konfiguration des DS in [Advanced] Mapping Rules bereits eine Einstellung diesbezüglich getätigt worden, besteht hier keine Auswahlmöglichkeit mehr.

Wird ein Attribut gemappt, so muss neben dem Tag auch noch das zu mappende Attribut ausgewählt werden. Dieses muss in der Vorlage bereits vorhanden sein.

Ist die Zuordnung erfolgreich getätigt worden, kann das Template gespeichert werden. Neben dem Titel des Templates muss auch noch die Art des Templates, page template oder content element, ausgewählt werden. Gespeichert werden Templates in einem SysFolder oder einer Seite.

4.5.4. Einbindung von Templates in eine Seite

TemplaVoila wird im Root-Template, also im obersten Template einer Webseite, durch folgende Zeilen eingebunden:

```
page = PAGE
page.typeNum = 0
page.10 = USER
page.10.userFunc = tx_templavoila_pil->main_page
```

In der ersten Zeile wird die aktuelle Seite `PAGE` der Variable `page` zugewiesen.

Die zweite Zeile bedeutet, dass die folgenden Anweisungen für alle Seiten gelten, die auf dieses Template zurückgreifen, also praktisch für jede Seite, die hierarchisch unter der Root-Seite liegt.

Die Zuweisung von `USER` an `page.10` bewirkt, dass in `page` an der Stelle `10` Inhalt mit Hilfe einer benutzerdefinierten Funktion generiert wird; die Funktion wird schließlich in Zeile vier zugewiesen.

Die Zuweisung eines page templates muss pro Webseite nur einmal erfolgen, nämlich in der Wurzelseite. Durch die Vererbung benutzen alle Unterseiten automatisch das gleiche Seitentemplate. Diese Zuweisung erfolgt in den Seiteneigenschaften im Reiter „Extended“. Hier muss sowohl die zu verwendende „Data Structure“ als auch das Template Object ausgewählt werden.

Wird nun eine neue Seite erstellt und im Reiter Extended das gewünschte Template ausgewählt, so werden alle im Seiteneditor bearbeitbaren Felder des Templates automatisch in den Seiteneditor integriert, wo Inhalte eingefügt werden können.

4.5.5. Aktualisierung eines Templates

Die einem Template zu Grunde liegende HTML-Vorlage kann auch geändert werden. TemplaVoila erkennt die Änderung und zeigt im Backend eine Warnung an, dass eben eine Vorlage geändert wurde und das Mapping gegebenenfalls zu aktualisieren ist. TemplaVoila versucht zwar, das Mapping an die Änderungen anzupassen, dies gelingt jedoch nur, wenn die Struktur des Vorlagendokuments nicht verändert wurde. Deswegen muss bei jeder Änderung der Vorlage ein berechtigter Backendbenutzer das Mapping überprüfen und gegebenenfalls neu durchführen bzw. aktualisieren. Erst wenn das Mapping vollständig ist, wird das neue Template als Vorlage benutzt.

4.6. Lokalisierung

Typo3 bietet vielfältige Möglichkeiten der Lokalisierung einer Webseite. Voraussetzung für die Mehrsprachigkeit ist die Definition einer zusätzlichen Sprache in der Wurzel des Seitenbaumes. Da eine Standardsprache, die „default-Sprache“, sowieso vorhanden ist, genügt die Erstellung einer zweiten Sprache im Seitenbaum, um die Mehrsprachigkeit zu aktivieren.

Anschließend kann man bereits die Übersetzungen vornehmen. Typo3 hat bezüglich Mehrsprachigkeit eine lange Evolution hinter sich. Dementsprechend gibt es einige Konzepte und auch Möglichkeiten, diese Konzepte zu kombinieren, was nicht immer den gewünschten Effekt haben muss.

Im Anschluss wird ein Überblick über die Konzepte und das Zusammenspiel dieser gegeben.

4.6.1. Übersetzung und Konfiguration einer Seite

In den Seiteneigenschaften lassen sich wichtige Felder einer Seite übersetzen. Es sind dies „Seitentitel“, „Untertitel“, „Navigationstitel“, „Abstract“, „Autor“, „Email des Autors“, „Schlüsselwörter“, „Beschreibung“ und „assoziierte Dateien“.

Im Reiter „Options“ der Seiteneigenschaften kann konfiguriert werden, dass die Seite versteckt wird, wenn sie in der im Frontend aktuell gewählten Sprache nicht verfügbar ist. Im Seitentemplate können weitere Optionen diesbezüglich konfiguriert werden. Bei Fehlen einer Übersetzung ist es entweder möglich, dass dies als Fehler ausgegeben wird, oder, dass die

nicht übersetzten Inhalte in der default-Sprache angezeigt werden. Die restlichen Elemente, wie Menüs etc. werden trotzdem, soweit vorhanden, in der übersetzten Sprachversion angezeigt.

4.6.2. Übersetzung der Standard-Elemente

Content Elements, also Elemente, die im Seiteneditor eingefügt werden können, wie Texte, Texte mit Bildern, usw. können ebenfalls übersetzt werden. Dabei werden nach Auswahl der Zweitsprache die Inhalte der default-Sprache unterhalb der Felder der neuen Sprache angezeigt. Dies soll für eine leichtere Übersetzbarkeit sorgen.

Content Elements können nicht nur in der default-Sprache und den hinzugefügten Sprachen erstellt werden, auch die Auswahl „all“ ist möglich, was bedeutet, dass dieses Content Element immer angezeigt wird, egal welche Sprache gerade aktiv ist.

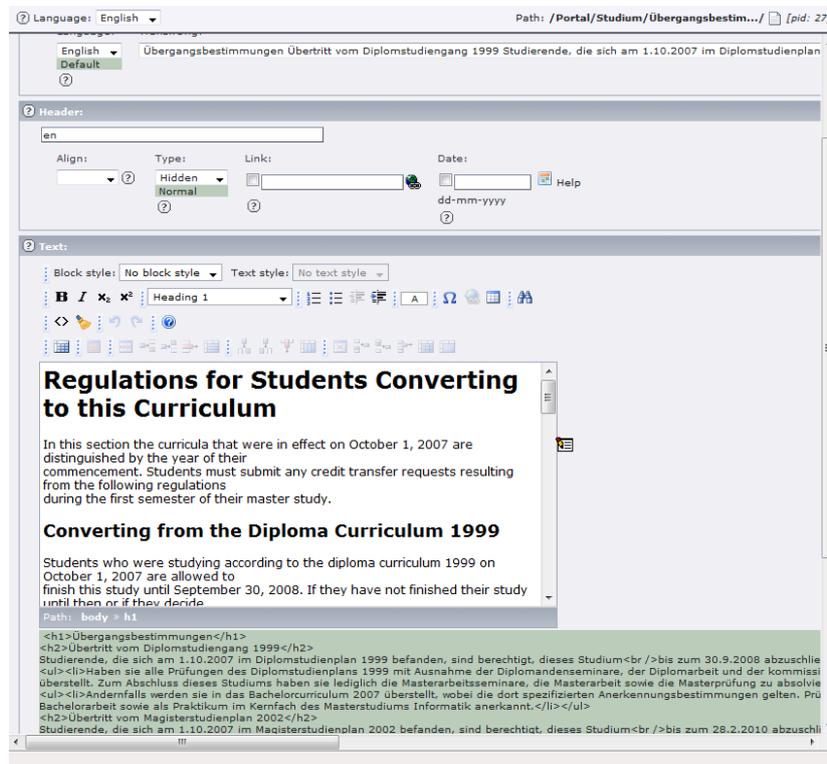


Abbildung 8: Übersetzung von Content Elements

Typo3 erlaubt verschiedene Einstellungen, wie Content Elements in verschiedenen Sprachen ausgegeben werden sollen. Die Konfiguration erfolgt mit Hilfe von TypoScript in einem Template.

`config.sys_language_overlay = 1`, bedeutet, dass alle Felder, unabhängig vom Übersetzungsstatus, angezeigt werden. Ist eine Übersetzung vorhanden, so wird diese angezeigt.

`config.sys_language_overlay = hide_Non_Translated`, bewirkt, dass auf einer Seite nur Elemente, welche in der aktuellen Sprache verfügbar sind bzw. die mit der Spracheinstellung „all“ angelegt wurden, angezeigt werden.

`config.sys_language_overlay = 0` wiederum bewirkt, dass unabhängig vom Übersetzungsstatus eines Feldes immer nur der Inhalt in der default-Sprache angezeigt wird.

Für Verwirrung sorgen kann die Tatsache, dass mit Hilfe von TemplaVoila erstellte Flexible Content Elements ebenfalls Content Elements sind, und dementsprechend auch übersetzt werden können.

TemplaVoila bietet jedoch ganz eigene Spracheinstellungen an, sodass es hier möglich ist, Elemente „doppelt“ zu übersetzen.

4.6.3. Spracheinstellungen von TemplaVoila

In TemplaVoila werden die Spracheinstellungen in der Data Structure, welche in XML-Form vorliegt, gespeichert.

Die relevanten Tags sind hierbei die beiden lang-Tags.

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<T3DataStructure>
  <meta type="array">
    <langChildren type="integer">1</langChildren>
    <langDisable type="integer">0</langDisable>
  </meta>
```

Von den vier Kombinationsmöglichkeiten der beiden Eigenschaften sind nur 1-0, 0-0 und x-1 relevant.

Bei der Einstellung von x-1 wird die Sprachunterstützung innerhalb von TemplaVoila deaktiviert. Flexible Content Elements müssen, wie oben bei Content Elements beschrieben, separat für jede Sprache angelegt werden.

1-0 aktiviert die Vererbung in TemplaVoila. Dies ist ein mächtiges Konzept, welches Übersetzungen strukturell sehr eng aneinander kettet. Wie in Abbildung 9 zu sehen, können im Seiteneditor direkt untereinander beide Sprachversionen konfiguriert werden. Zu beachten ist, dass beide Sprachversionen von TemplaVoila der default-Sprache des Content Elements zugeordnet sind.

Würde man die Content Element-Sprache ändern, so wären alle Felder noch einmal leer vorhanden. Alle Felder sind also für beide Sprachen vorhanden, relevant sind jedoch nur die Inhalte der Felder der default-Sprache des Content Elements. Die TemplaVoila-Einstellungen überschreiben die Spracheinstellungen des Content Elements.

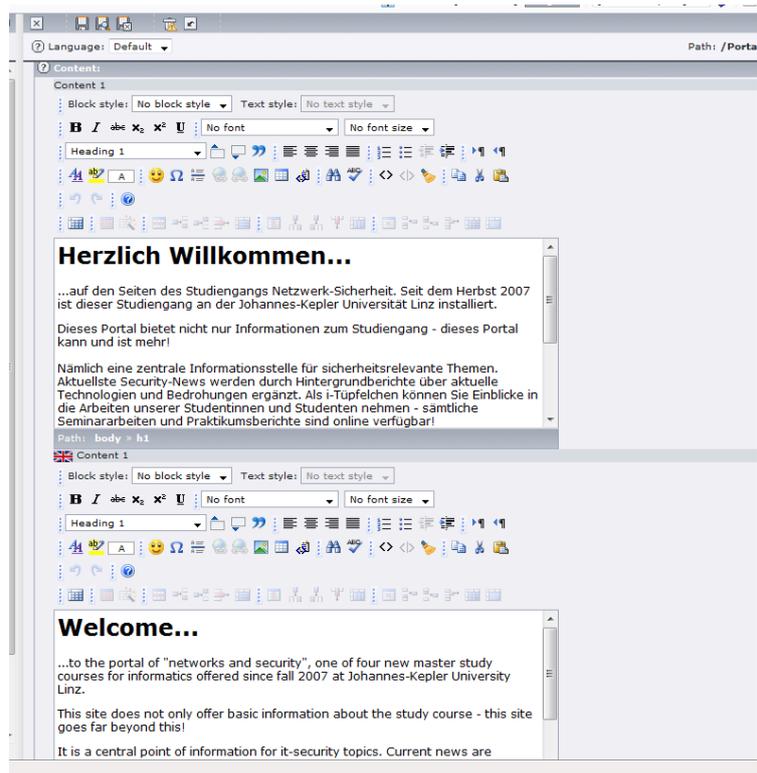


Abbildung 9: Sprachvererbung in TemplaVoila

Besonders interessant ist, dass Flexible Content Elements wiederum Content Elements enthalten können. Dies können z.B. Plugins, weitere Flexible Content Elements, etc. sein. Problematisch ist hierbei, dass in ein Flexible Content Element nur Referenzen auf bereits bestehende Plugins eingetragen werden können, was wiederum bedeutet, dass man das Plugin der anderen Sprachversion bereits vorher anlegen muss. Dies geschieht tatsächlich in der Alternativsprache des Content Elements; anschließend muss das Plugin im Editor des Flexible Content Elements referenziert werden.



Abbildung 10: Verschiedene Sprachversionen eines Plugins

Die Möglichkeit 0-0 bewirkt, dass die Sprachfunktionen von TemplaVoila zwar aktiviert sind, die Vererbung jedoch nicht aktiv ist. Die Übersetzungsmöglichkeit ist zwar vorhanden, die verschiedenen Sprachversionen sind aber nicht eng aneinander gebunden, was strukturell unterschiedliche Sprachversionen ermöglicht. Dies äußert sich auch darin, dass im Flexible

Content Editor zuerst alle Felder der einen Sprache und erst danach die Felder der anderen Sprache angeordnet sind. Ein Vorteil dieser Technik ist, dass unterschiedliche Seiten (auch strukturell) für unterschiedliche Sprachen angelegt werden können. [SKÅ-06/1, PÖT-06]

4.6.4. Übersetzung von Extensions

Auch Extensions, die Erweiterungen von Typo3, können mehrsprachig gestaltet werden. Weiterführende Informationen dazu sind im Abschnitt 4.7.7 zu finden.

4.7. Extensions – die Erweiterungen in Typo3

Eine der großen Stärken von Typo3 ist die Erweiterbarkeit. Ob Plugins für das Frontend oder Module für das Backend, praktisch jede Funktionalität ist in Typo3 mit Erweiterungen realisierbar. In diesem Kapitel werden die verschiedenen Arten von Erweiterungen präsentiert, sowie auf deren Aufbau und Konzeption eingegangen. Weiters werden das Typo3 Extension Repository und die Erweiterung „kickstarter“ vorgestellt.

4.7.1. Grundsätzliche Eigenschaften von Extensions

Extensions zeichnen sich in Typo3 durch verschiedene, grundlegende Eigenschaften aus:

- Alle Dateien einer Erweiterung befinden sich in einem Ordner.
- Erweiterungen können leicht installiert, geändert und ausgetauscht werden.
- Jede Erweiterung besitzt einen sogenannten „Extension-Key“. Dieser identifiziert die Erweiterung (im Idealfall) weltweit eindeutig und wird außerdem zur Benennung von Dateien, Tabellen, etc. benutzt.
- Eine Erweiterung kann durch die „Typo3 Core API“ in die Funktionen des Kerns eingreifen und diese erweitern.
- Erweiterungen können in das „Typo3 Extension Repository“ (TER) hochgeladen werden und stehen damit anderen Entwicklern zur Verfügung. Im TER kann eine Extension nur mit einem eindeutigen Extension-Key hinzugefügt werden.
- Erweiterungen stehen im Format *.t3x* zur Verfügung.

4.7.2. Arten von Extensions

Extensions lassen sich in Typo3 sowohl nach ihrem Installationsort als auch nach ihrer Funktionalität klassifizieren.

4.7.2.1. Einteilung nach Installationsort

Je nach dem Installationsordner können in Typo3 folgende Extensions unterschieden werden:

- *Globale Extensions*

Diese Erweiterungen werden in *typo3/ext/* gespeichert, also im Quellcode-Ordner. Sie sind daher für alle Typo3-Webseiten auf einem Server verfüg- und nutzbar, die auf den gleichen Source-Code zurückgreifen. Obwohl globale Erweiterungen im Source-Code Ordner gespeichert werden, müssen sie doch nicht mit ihm gemeinsam ausgeliefert werden.

- *Systemextensions*

Wie auch globale Extensions werden diese direkt im Source-Code Ordner gespeichert, und zwar im Ordner *typo3/sysex/*.

Systemerweiterungen sind automatisch bei Distributionen des Source-Codes dabei. Außerdem werden sie gemeinsam mit dem Systemkern gewartet und erweitert, sodass man diese Erweiterungen nicht manuell aktualisieren muss.

- *Lokale Erweiterungen*

Diese Erweiterungen befinden sich im Ordner *typo3conf/ext/*. Sie sind nur für das jeweilige Projekt verfügbar und müssen manuell aktualisiert werden. Benutzererstellte Erweiterungen werden ebenfalls als lokale Erweiterungen installiert.

Befindet sich die gleiche Erweiterung in zwei oder allen drei Installationsordnern, so gilt folgende Priorität: Lokal > Global > System.

4.7.2.2. Einteilung nach Funktionalität

Je nach zukünftiger Funktion einer Erweiterung kann sie bei der Erstellung einer bestimmten Kategorie zugeordnet werden. Folgende Kategorien sind in Typo3 verfügbar:

- *Backend*

Ist eine beliebige Erweiterung für das Backend, jedoch kein Modul.

- *Backend-Modul*

Wie Backend, allerdings mit eigener Ausgabe und GUI im Backend.

- *Frontend*

Fügt Funktionalität zum Frontend hinzu, ohne jedoch ein echtes Plugin zu sein.

- *Frontend-Plugin*
Fügt ein Frontend-Plugin hinzu; mit Ausgabe bzw. Benutzerinteraktionsmöglichkeit.
- *Services*
Enthält Service-Klassen, welche von anderen Extensions benutzt werden können.
- *Templates*
Enthält Templatedateien für Webseiten.
- *Example*
Ist ein Beispiel zu einer anderen Erweiterung.
- *Documentation*
Kann z.B ein Tutorial zu einer Erweiterung sein.
- *Miscellaneous*
Alles andere.

Die Einteilung in die obigen Kategorien hat im praktischen Arbeiten keine Auswirkungen auf die Funktionsweise oder die Möglichkeiten einer Extension, sie dient einzig der leichteren Klassifizierung einer Erweiterung. [SKA-07/2]

4.7.3. Die Verwaltung von Extensions

Erweiterungen werden in Typo3 in einem Modul im Backend, dem „Extension Manager“ verwaltet.

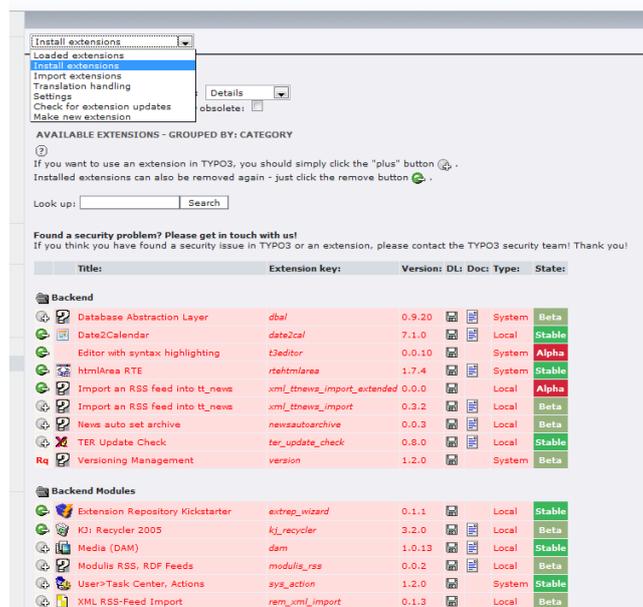


Abbildung 11: Extension Manager im Typo3 Backend

Dieser ermöglicht sowohl den Download, die Installation als auch die grundlegende Konfiguration von Erweiterungen.

Im Folgenden werden die verschiedenen Funktionen erläutert:

4.7.3.1. Import extensions

Typo3 ermöglicht den direkten Download von Extensions aus dem Typo3 Extension Repository (TER). Gesucht werden kann nach dem Extension Key, welcher im TER eindeutig sein muss. Ist die Extension im TER verfügbar, kann sie mit einem Mausklick direkt heruntergeladen und anschließend installiert werden.

4.7.3.2. Install extensions/Loaded Extensions

Wie in Abbildung 11 zu sehen, gibt „Install Extensions“ einen Überblick über alle im System befindlichen Extensions, sortiert nach den im Punkt 4.7.2.2 vorgestellten Kategorien. Installierte Extensions, erkennbar am grünen „Minus“, können per Mausklick deinstalliert werden, ungeladene Erweiterungen, erkennbar am grauen „Plus“, können installiert werden. Erweiterungen, welche für den Betrieb von Typo3 erforderlich sind, sind hier mit „Rq“ gekennzeichnet.

„Loaded Extensions“ zeigt nur die geladenen Extensions, während Install extensions alle auf dem System verfügbaren anzeigt.

4.7.3.3. Translation handling

Die Mehrsprachigkeit von Typo3 beschränkt sich nicht nur auf die Webseiten und das Backend, auch für viele Extensions existieren zusätzliche Sprachdateien online. Der Extension Manager erlaubt deshalb auch die Suche nach Sprachdateien und ihre Einbindung in die vorhandenen Extensions.

4.7.3.4. Check for extension updates

Ab der Version 4.2 ist es möglich, direkt im Extension Manager die vorhandenen Extensions auf Updates zu überprüfen und diese gegebenenfalls herunterzuladen und zu installieren.

4.7.3.5. Settings

Hier lassen sich grundlegende Einstellungen des Extension Managers, wie die verwendeten Download-Server oder die Login-Daten für den TER (nur zum Upload von Extensions erforderlich) konfigurieren.

Es besteht auch die Möglichkeit, die Suche nach Extensions nur auf sichere, also begutachtete Erweiterungen zu beschränken.

4.7.4. Struktur einer Extension

Wie weiter oben schon erwähnt, besitzt jede Extension einen (im Idealfall) weltweit eindeutigen Schlüssel, den Extension Key. Alle Dateien einer Extension müssen sich innerhalb eines nach dem Extension Key benannten Ordners befinden.

Innerhalb dieses Ordners sollen Dateien für Frontend-Plugins im Unterordner */pi1/* (bzw. für mehrere Plugins *pi1, pi2, etc...*), Dateien für Backendmodule im Unterordner */mod1/* und Hilfsdateien im Unterordner */res/* gespeichert werden.

Es gibt für Extensions einige reservierte Dateinamen. Als Konvention sind diese immer mit dem Präfix „*ext_*“ benannt.

Reservierte Dateinamen sind:

- *ext_emconf.php*
Enthält Metainformationen zur Extension wie Name, Status, Kategorie, Abhängigkeit von anderen Extensions und einiges andere mehr. Diese Datei muss vorhanden sein, sonst erkennt der Extension Manager die Erweiterung nicht.
- *ext_localconf.php*
Erweitert die *localconf.php* und kann zusätzliche Konfiguration und PHP-Klassen enthalten.
- *ext_tables.php*
Enthält Konfiguration zu Modulen, Tabellen,...
- *ext_tables.sql, ext_tables_static + adt.sql*
Die von der Extension benötigte Tabellenstruktur wird hier im SQL-Format gespeichert bzw. können auch statische Einträge definiert werden.
- *ext_conf_template.txt*
Einstellungen der Extension, welche im Extension Manager konfigurierbar sein sollen, können mit Hilfe dieser Datei in TypoScript-Syntax angegeben werden.
- *ext_icon.gif*
Zur leichteren Erkennbarkeit kann eine Extension auch ein eigenes Symbol haben.
- *locallang.php*
Hier können für Texte, Labels etc. in der Extension verschiedene Sprachversionen angelegt werden. Sind Übersetzungen für die Extension in der aktuell ausgewählten Sprache vorhanden, so wird von Typo3 automatisch die richtige Sprachversion eingesetzt.

- *class.ext_update.php*

Extension-Entwickler haben die Möglichkeit, eine eigene Update-Prozedur für ihre Erweiterungen zu integrieren. Dies wird durch die Implementierung der Klasse `ext_update.php` gewährleistet.

- *ext_api.php.dat*

Enthält ein serialisiertes PHP-Array mit API-Informationen für die PHP-Klassen der Erweiterung. [SKA-07/3]

4.7.5. Erstellung einer Extension mit Hilfe von „Kickstarter“

Der Kickstarter ist, wie der Name schon sagt, kein Editor und kein Entwicklungswerkzeug, er nimmt dem Extension-Entwickler lediglich die Arbeit der Erstellung der Standard-Dateien, sozusagen des „Gerüsts“ der Extension ab und bietet dabei folgenden Einstellungsmöglichkeiten:

- *Extension key*

Am Anfang muss ein „Extension Key“ vergeben werden. Dieser sollte möglichst eindeutig und aussagekräftig sein. Soll die Extension in das Typo3 Extension Repository (TER) hochgeladen werden, ist die Eindeutigkeit des Extension Keys ein Muss.

- *General info*

Hier können Metainformationen zur Extension eingetragen werden. Autornamen und -email, sowie Status und Art der Extension, Abhängigkeiten von anderen Extensions sowie Titel und Beschreibung der Erweiterung können eingetragen werden. Diese Informationen sind später sowohl im TER als auch im Extension Manager verfügbar. Die unter „Abhängigkeiten“ eingetragenen Extensions müssen vor Installation selbst installiert sein bzw. wird das von Typo3 automatisch erledigt.

- *New Database Table*

Die bei der Installation anzulegenden Datenbanktabellen werden unter diesem Punkt erstellt. „Field name“, „Field title“ sowie eine Übersetzung des Titels können ebenso vorgenommen werden wie die Einstellung „Feldtyp“. Für die anzulegende Tabelle können neben den eigentlichen Datenbankeinstellungen noch Einstellungen zur Behandlung im Typo3-Backend getätigt werden, z.B. kann die Versionierung der Tabelle aktiviert werden, oder auch die Eigenschaft, Tabelleninhalte mehrsprachig zu erstellen.

- *Extend existing Table*

Auch bestehende Tabellen können erweitert werden. Dazu ist neben der Angabe der zu erweiternden Tabelle auch die Information zu den Tabellenfeldern nötig.

- *Backend Modules*

Hier muss die Information angegeben werden, ob der Zugriff auf das Modul über einen Haupt- oder Submenüpunkt erfolgen, wo es im Backend angezeigt werden soll und ob es nur dem Administrator zur Verfügung steht.

Name, Beschreibung und Kurzbeschreibung können ebenfalls angegeben werden.

- *Integrate in existing Modules*

Erlaubt die Integration eines Backendmodules in die Module Web-Func, -Info, -Template bzw. User-Taskcenter.

- *Clickmenu items*

Die kontext-sensitiven Menüs in Typo3 können mit eigenen Menüpunkten bzw. Funktionen erweitert werden. Dabei kann auch ein „second-level“-Menü aufgerufen werden. Weiters ist die Beschränkung auf bestimmte Elementtypen möglich. Auch das Löschen von Menüpunkten ist erlaubt.

- *Services*

Services erlaubt die Erstellung von Hilfsklassen. Diese können die Funktionalität von Typo3 bzw. von Extensions erweitern, ohne, dass der Quellcode geändert werden muss. Benötigt das Service externe Programme, so kann dies ebenso angegeben werden wie eventuelle Betriebssystemabhängigkeit.

- *Static TypoScript code*

Erlaubt die Angabe von statischem, nicht änderbarem TypoScript-Code.

- *TSconfig*

Erlaubt die Angabe von Page TSconfig bzw. User TSconfig. Mit Hilfe dieses Features können Einstellungen einer Extension mit Standard-Werten vorbelegt werden. Nachträglich können die Werte jedoch trotzdem geändert werden.

- *Setup languages*

Hier können die in der Extension verfügbaren Sprachen definiert werden.

Mit einem Klick auf „TotalForm“ erhält man alle Konfigurationsformulare sequentiell präsentiert, sodass man einen schnellen Überblick über die bisher getroffenen Einstellungen bekommt.

„View result“ zeigt die von Kickstarter produzierten Dateien sequentiell an. Diese können auch gleich ins lokale Extension Repository geschrieben und installiert werden.

Mit Hilfe von „D/L as file“ wird die mit dem Kickstarter erstellte Extension als *.t3x*-Datei heruntergeladen.

Anschließend an die Erstellung des Gerüsts muss die eigentliche PHP-Programmlogik geschrieben werden, wobei die von Typo3 benötigten Klassen und auch die aufgerufenen Methoden inklusive Basisinitialisierung bereits vorhanden sind, sodass sich der Entwickler voll auf die Erstellung der Programmlogik konzentrieren kann.

4.7.6. Erstellung der Programmlogik

Die Programmdateien einer Extension liegen je nach der Art der Extension in unterschiedlichen Ordnern.

Für Frontend-Plugins liegen die zum Rendern benutzten PHP-Dateien im Unterordner */pi/*. Die Haupt-PHP-Datei heißt `class_<extension-key>_pi<Nr. des Plugins>.php` und enthält die `main`-Methode mit einigen Initialisierungen und der Ausgabe von Dummy-Inhalt.

Jede Klasse eines Frontend-Plugins ist von der Basisklasse `tslib_pibase` abgeleitet. Diese enthält notwendige und praktische Funktionen, die zum Beispiel zur Lokalisierung der Extension gebraucht werden.

Die Funktion `main` besitzt die Parameter `$content` und `$conf`. Der Parameter `$content` enthält die Ausgabe des Plugins als String, in `$conf` sind die Einstellungen des Plugins gespeichert. Von den Initialisierungen ist besonders `$this->pi_loadLL()`; wichtig. Diese Zeile lädt die Sprachdatei einer Extension und macht damit die Lokalisierung möglich. Wie genau die Lokalisierung vor sich geht, wird weiter unten erläutert.

Anschließend kann der Inhalt der Variable `$content` verändert und angepasst werden. Frontend-Plugins werden in die Webseite eingebettet und geben daher meist HTML-Codestücke zurück.

Die letzte Zeile der `main`-Methode fügt die Inhalte des Plugins in die Seite an der gewünschten Stelle ein.

```
require_once(PATH_tslib.'class.tslib_pibase.php');

class tx_ktstudentfilesystem_pi1 extends tslib_pibase {
    function main($content,$conf)    {
        $this->conf=$conf;
        $this->pi_setPiVarDefaults();
        $this->pi_loadLL();
        ...
        return $this->pi_wrapInBaseClass($content);
    }
}
```

```

if (defined('TYPO3_MODE') &&
$TYPO3_CONF_VARS[TYPO3_MODE]['XCLASS']['ext/kt_student_file_system/pi1/class.tx_
ktstudentfilesystem_pi1.php']) {
    include_once($TYPO3_CONF_VARS[TYPO3_MODE]['XCLASS']['ext/kt_student_file_syst
em/pi1/class.tx_ktstudentfilesystem_pi1.php']);
}

```

Die letzten Zeilen binden die Klasse in die Abläufe von Typo3 und verhindern eine Ausführung außerhalb dieses Kontexts.

4.7.7. Mehrsprachigkeit in Extensions

Mit der oben beschriebenen Funktion `$this->pi_loadLL()`; wird die zu einer Extension gehörige Sprachdatei geladen. Diese Sprachdatei, die `locallang.xml`, liegt üblicherweise im selben Verzeichnis wie die Klasse, die sie benutzt.

Wie der Dateiname schon sagt, handelt es sich bei der Sprachdatei um eine XML-Datei, in der die verschiedenen Übersetzungen abgespeichert sind. Neben den Metainformationen, wie Typ und Beschreibung, werden die verschiedenen Sprachen über ihren „languageKey“ identifiziert und die Übersetzungen in Tags mit der Bezeichnung „label“ gespeichert. Der Name des zu ersetzenden Tags ist dabei als Attribut „index“ abgespeichert. Es lassen sich beliebig viele Sprachen mit beliebig vielen Labels hinzufügen.

`locallang.xml` sieht im Konkreten wie folgt aus:

```

<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<T3locallang>
    <meta type="array">
        <type>module</type>
        <description>Language labels for extension</description>
    </meta>
    <data type="array">
        <languageKey index="en" type="array">
            <label index="description">english text</label>
        </languageKey>
        <languageKey index="default" type="array">
            <label index="description">Deutscher Text</label>
        </languageKey>
    </data>
</T3locallang>

```

Man sieht – die Standardsprache („default“) ist Deutsch, Englisch ist die Zweitsprache.

Die Einbindung der Strings in die Ausgabe passiert direkt am gewünschten Platz in der Klasse mit Hilfe der Funktion `$this->pi_getLL('description')`.

Automatisch wird dann beim Rendern der Ausgabe des Plugins die aktuelle Sprache erkannt und der korrekte Label-Inhalt eingesetzt.

4.7.8. Die Verwendung von Ajax in Extensions

„Ajax“ ist eine Webtechnologie, mit der Inhalte von Webseiten nachgeladen und aktualisiert werden können, ohne, dass die Webseite neu geladen werden muss. Dies minimiert einerseits die Datenübertragung und sorgt andererseits dafür, dass speicherintensive Inhalte wie Bilder und Videos im Hintergrund nachgeladen werden können, während die restliche Webseite schnell vollständig verfügbar ist.

Ajax kombiniert JavaScript-Funktionen mit serverseitigen Skripts, um die gewünschten Inhalte an der gewünschten Stelle einzufügen. Prinzipiell ist es daher möglich, Ajax auch in Typo3 zu realisieren, ohne dabei auf Funktionen von Typo3 zurückzugreifen. Dies kann einfach geschehen, indem mit JavaScript ein externes PHP-Skript aufgerufen wird, welches die gewünschten Inhalte liefert. Allerdings verliert man dadurch die Vorteile, die die Typo3-Umgebung bietet - die Datenbankfunktionen genauso wie die Funktionen zur Prüfung der Benutzerberechtigung im Frontend und auch die Lokalisierungsfunktionen.

Typo3 bietet daher ein Konzept für Extensions, um Daten per Ajax zu laden: das Konzept der „eID“.

4.7.8.1. Ajax-Request mit Hilfe von JavaScript

Wie oben erwähnt, wird zum Absetzen eines „Ajax-Requests“ eine entsprechende JavaScript-Bibliothek benötigt. Im konkreten Fall wird dazu die in Typo3 integrierte „Prototype-Bibliothek“ verwendet.

Um einen Request abzusetzen, muss ein neues Objekt `Ajax.Updater(container, url[, options])` oder `Ajax.Request(url[, options])` mit den benötigten Parametern instanziiert werden.

`Ajax.Request` erlaubt neben der aufzurufenden URL noch die Angabe von weiteren Optionen. Neben Angaben über die Parameter kann auch die Auswahl getroffen werden, ob die Parameter über GET oder POST übertragen werden sollen.

Optional kann auch mit `onComplete:<Funktionsname>` eine Funktion definiert werden, welche bei Abschluss des Requests mit dem Response als Eingabeparameter aufgerufen wird.

`Ajax.Updater` benötigt zusätzlich noch die Angabe eines Containers. Das ist die ID jenes HTML-Elements, in das die Antwort direkt ausgegeben werden soll. Auch dem Updater können optional Parameter und Zusatzeinstellungen übergeben werden.

4.7.8.2. Request in Typo3

Damit eine Extension die Ajax-Funktionen von Typo3 nutzen kann, muss in der Datei *ext_localconf.php* ein eID-Schlüsselwort sowie das aufzurufende PHP-Skript, welches innerhalb des Extensionordners liegen muss, definiert werden.

```
<?php
$TYPO3_CONF_VARS['FE']['eID_include']['eidKey'] = 'EXT:extKey/script.php';
?>
```

Um die eID-Funktionalität nutzen zu können, muss die benutzte Extension ihren Request direkt an *index.php* stellen. Als einer der Parameter muss die eID der Extension angegeben werden. Zu Beginn des Renderingprozesses der *index.php* wird überprüft, ob der Parameter eID vorhanden ist. Ist dies der Fall, so wird der Renderingprozess abgebrochen und stattdessen das in der *ext_localconf.php* angegebene Skript ausgeführt. Die von der *index.php* retournierten Inhalte stammen in diesem Fall ausschließlich vom benutzerdefinierten Skript.

Auf Grund des frühen Abbruchs des Renderingprozesses stehen nicht alle Typo3-Funktionen zur Verfügung. Speziell für den eID-Mechanismus existieren daher Datenbankfunktionen sowie die Funktionen zur Frontend-Benutzerberechtigung.

```
<?php
    $feUserObj = tslib_eidtools::initFeUser();
    tslib_eidtools::connectDB();
    ...
?>
```

Auch Mehrsprachigkeit lässt sich mit eID realisieren, was allerdings nicht ganz einfach ist.

Das Ergebnis des Requests kann mittels `print`-Befehl an das aufrufende JavaScript zurückgegeben werden und wird dort je nach der aufrufenden Methode behandelt und in das Dokument eingefügt.

5. *security.fim.uni-linz.ac.at* – Konzept

In diesem Kapitel wird auf das strukturelle und inhaltliche Konzept von *security.fim.uni-linz.ac.at* eingegangen.

5.1. Gliederung von *security.fim.uni-linz.ac.at*

Das Portal „Netzwerke und Sicherheit“ bietet Informationen sowohl zum Studium an sich, als auch fachbezogene Inhalte im Kontext von „Netzwerke und Sicherheit“. Die Webseite enthält daher neben dem Hauptportal, der Einstiegsseite mit Begrüßungsworten und einem ersten

Nachrichtenüberblick, auch verschiedene Unterportale, welche die Aufteilung in studienrelevante bzw. fachliche Informationen widerspiegelt. Auf Grund der thematischen Breite des Studiums werden auch die fachlichen Informationen in einzelne Portale gegliedert, wobei mit dem Launch des Portals nur die Unterscheidung zwischen „Sicherheit“ und „Forensik“ getroffen wird.

Dies ergibt eine Gliederung von *security.fim.uni-linz.ac.at* in vier Teile, deren Inhalt im Folgenden genauer beschrieben wird.

5.1.1. **Home**

Das Hauptportal von *security.fim.uni-linz.ac.at* enthält neben einem Begrüßungstext zwei Bereiche mit Nachrichten. Einerseits sind dies die aggregierten, themenbezogenen News sämtlicher, in der gewählten Sprache vorhandenen Quellen, andererseits sind dies die von Redakteuren erstellten Neuigkeiten, die das Studium direkt betreffen. Im vierten Block der Startseite werden die zuletzt auf das Portal hochgeladenen Dateien angezeigt.

So soll sichergestellt werden, dass ein Besucher des Hauptportals mit einem Blick umfassend über alle relevanten neuen Entwicklungen im Studium, im Fachkontext, als auch im Hinblick auf das Portal selbst informiert ist.

Im Rahmen des Hauptportals werden dem Benutzer auch Kontaktinformationen, das Impressum, eine Suchfunktion sowie die Sitemap angeboten.

5.1.2. **Studium**

Informationen zum Studium sind ein essentieller Bereich von *security.fim.uni-linz.ac.at*; soll die Plattform doch auch als Werbeträger und Informationsquelle für interessierte Bakkalaureatsstudentinnen und -studenten dienen. Wichtig ist daher, im Unterportal „Studium“ so umfassend und interessant als möglich zu informieren. Auf der Hauptseite dieses Unterportals werden neben einem kurzen Einleitungstext daher die studienrelevanten Nachrichten präsentiert.

Neben den „Hard Facts“ zum Studium, wie Voraussetzungen, Studienplan und Übergangsbestimmungen, sollen hier auch „Soft Facts“, wie die Antwort auf die Frage „Was bringt mir das Studium?“ vermittelt werden. Außerdem werden Informationen zu den Lehrveranstaltungen sowie auch zu den Lehrenden angeboten. Abgerundet wird das Angebot durch die Vorstellung der für die Studenten relevanten Organisationen „Österreichische Hochschülerschaft“ (ÖH) und „Studienkommission Informatik“ (Stuko).

5.1.3. **Sicherheit**

Neben den dem Sicherheitsportal zugeordneten Fächern werden Artikel und Hintergrundberichte angeboten. Außerdem wird als besonderes Highlight das 10h Praktikum von Karl Thier, „Der gläserne Nutzer“ angeboten. Dieses zeigt, wie leicht Nutzer mit Hilfe von Technologien wie JavaScript, CSS und HTML ausspioniert, bzw. welche Informationen über einen Besucher einer Webseite direkt eruiert werden können.

5.1.4. **Forensik**

Neben dem Fach IT-Recht und Computerforensik werden noch Artikel und Hintergrundberichte zum Thema angeboten. Ein „Fall der Woche“ rundet das Thema ab.

5.2. **Design von *security.fim.uni-linz.ac.at***

Das Design von *security.fim.uni-linz.ac.at* gliedert sich in folgende vier Teile:

5.2.1. **Header**

Neben grafischen Elementen, wie dem Logo und dem Namen des Studienganges befinden sich hier auch die Links zu Impressum, Suche, Kontaktformular und Sitemap. Außerdem ist ein Suchfeld untergebracht. Des Weiteren befinden sich die Hauptmenüelemente am unteren Rand des oberen Balkens, deren Untermenüpunkte sich durch Berührung mit dem Mauszeiger ausklappen lassen. Die Anordnung der Hauptmenüpunkte ermöglicht eine grafische Verbindung des ausgewählten Menüpunktes mit dem direkt darunter befindlichen Inhaltsbereich.

5.2.2. **Auswahlbereich**

Der Auswahlbereich links enthält ein Menü in Dropdown-Form zur Schnellauswahl einer Seite. Darunter werden die Untermenüpunkte des aktuell ausgewählten Hauptmenüpunktes angezeigt. Dies soll eine schnelle und unkomplizierte Navigation ermöglichen. Unter diesem Menü befindet sich schließlich der Bereich mit den Security-Widgets, die interessante Informationen in Kurzform bieten.

5.2.3. **Hauptinhaltsbereich**

Rechts des Auswahlbereiches unter dem Header befindet sich der Hauptinhaltsbereich. Hier werden, unterhalb der ebenfalls hier angezeigten „Breadcrumb“, die Inhalte der einzelnen

Seiten angezeigt, der Inhalt dieses Bereiches variiert daher von Seite zu Seite. Aus Gründen der Übersichtlichkeit wird für jedes Unterportal eine eigene Hintergrundgrafik angezeigt.

Prinzipiell werden im Hauptinhaltsbereich die Informationen sequentiell, sprich untereinander, angezeigt. Ausnahmen stellen die Hauptseiten von Haupt- bzw. Unterportalen dar. Diese sind prinzipiell fünfgeteilt. Eine Spalte oben zur textuellen Beschreibung des Portals bzw. für Begrüßungsworte; darunter zweimal zwei Spalten für seitenspezifische Inhalte wie News, aktuelle Dateien und ähnliches.

5.2.4. Fußzeile

Die Fußzeile bildet den Abschluss der Seite. Neben einem Copyright-Hinweis befinden sich hier auch Links zu interessanten Seiten sowie die Links zur Validierung der Konformität der Webseite mit XHTML-Strict bzw. CSS Version 2.



Abbildung 12: Seitenstruktur von *security.fim.uni-linz.ac.at*

6. Implementierung

In diesem Kapitel wird die Implementierung des Portals „Netzwerke und Sicherheit“ auf Basis des Content Management Systems „Typo3“ vorgestellt.

Ausgehend von der Seitenstruktur, der anschließenden Templateerstellung und –integration mit Hilfe von TemplaVoila und dem Anlegen der Inhalte soll die Entstehung Schritt für Schritt skizziert werden. Die dazu verwendeten Extensions werden vorgestellt und ihre Funktionsweise und Konfiguration erläutert.

6.1. Seitenstruktur

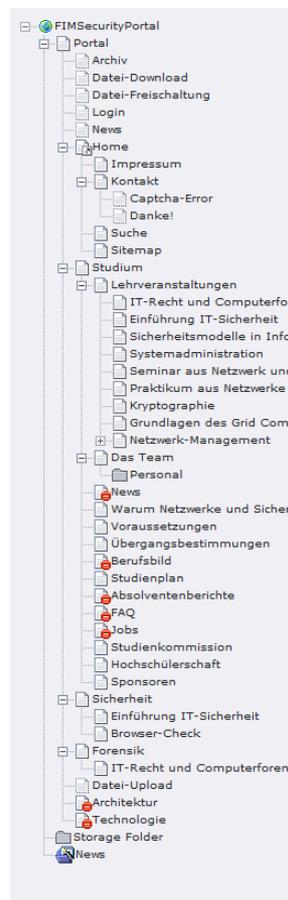


Abbildung 13: Seitenbaum in Typo3

Mit der Seitenstruktur im Seitenbaum wird die Erstellung von *security.fim.uni-linz.ac.at* begonnen. Eine Ebene unterhalb der Wurzel des Baumes „FIMSecurityPortal“ befindet sich die Einstiegs- bzw. Hauptseite des Portals, unter welcher sich alle anderen Seiten befinden. Wie in Abbildung 13 erkennbar, gibt es im gesamten Seitenbaum drei zusätzliche Ordner: Personal, Storage Folder und News. Auf die Funktion von Personal wird später noch eingegangen. Der Storage Folder enthält die Konfiguration der Seite in Form von

verschachtelten Templates und zusätzlich alle mit TemplaVoila generierten Seitenvorlagen bzw. Vorlagen für Inhaltselemente. In News werden alle von der Extension „tt_news“ verwalteten Nachrichten abgespeichert. Die Extension tt_news und ihr Einsatz in *security.fim.uni-linz.ac.at* werden ebenfalls später noch genauer behandelt.

6.2. Konfiguration – das Root-Template

Wie weiter oben erwähnt, benötigt jede Seite in Typo3 ein Template, welches die Seitenkonfiguration enthält. Das Haupttemplate wird in der Rootseite „Portal“ erstellt und automatisch auf alle anderen Seiten vererbt. Aus Gründen der besseren Lesbarkeit und Änderbarkeit wird in diesem Template jedoch keine Konfiguration vorgenommen, sondern nur ein anderes Root-Template eingebunden. Dieses liegt im Ordner „Storage Folder“ und enthält grundlegende Seiteneinstellungen.

```
page = PAGE
page.typeNum = 0
PAGE_TARGET=
page.meta.language = de
page.meta.author = Karl Thier
page.meta.description = Portal Studiengang Netzwerke und Sicherheit
page.meta.keywords = Sicherheit, Security, Networking, Netzwerke, JKU,
Johannes Kepler, Universität
```

Abhängig vom gerade aktuellen Portal wird eine spezifische Hintergrundgrafik angezeigt. Im folgenden Codestück wird in die HTML-Vorlage an der Stelle `coverimage` der Pfad zur Grafik eingefügt. Dieser Pfad ist in der Variable `bgimage` gespeichert, welche im Template des jeweiligen Unterportals definiert ist und hier dynamisch ersetzt wird.

```
bg = TEXT
bg.value = {$bgimage}
lib.coverimage < bg
```

Die Sprachkonfiguration ist ein wesentlicher Teil in Typo3. Folgendes Codestück definiert Deutsch als Standardsprache (`language_uid = 0`) und Englisch als Zweitsprache. `GP:L` definiert hier die GET- bzw. POST-Variablen. Die Auswahl der Sprache wird also einfach durch einen zusätzlichen Parameter `L` in der Seiten-URL getroffen.

```
# Deutsche Sprache, sys_language.uid = 0
config.linkVars = L
config.sys_language_uid = 0
config.language = de
config.locale_all = de_DE

# English language, sys_language.uid = 1
[globalVar = GP:L=1]
```

```
config.sys_language_uid = 1
config.language = en
config.locale_all = en_EN
[global]
```

Einbindung des Stylesheets

```
page.includeCSS.mainStyleSheet = fileadmin/templates/mainstyles.css
```

Folgender Code konfiguriert Typo3 zur XHTML-konformen Ausgabe des Quellcodes:

```
# Enable XHTML
config {
    doctype = xhtml_strict
    xhtml_cleaning = all
    htmlTag_setParams+xmlns="http://www.w3.org/1999/xhtml" xml:lang="de"
    lang = config.language
}
```

Typo3 bietet ein Feature zum Spam-Schutz von Email-Adressen, welches mit folgendem Code aktiviert wird:

```
config.pageTitleFirst = 1
config.spamProtectEmailAddresses = 1
config.spamProtectEmailAddresses_atSubst = @<span style="display: none;">dont-want-spam.</span>
```

Mit den folgenden Zeilen wird TemplaVoila als Templating-Engine in die Seite eingebunden:

```
page.10 = USER
page.10.userFunc = tx_templavoila_pil->main_page
```

Folgende Zeilen binden zwei Extensions in die Seite ein.:

```
lib.dates < plugin.tx_ktkusssdateexport_pil
lib.widgets < plugin.tx_ktsecuritywidget_pil
```

Auch die benötigten Konfigurationstemplates der einzelnen Extensions werden im Storage Folder gespeichert und in das Root-Template eingebunden.

Dieser Ansatz wurde gewählt, um die Konfigurationseinstellungen zu kapseln und dadurch leichter lesbar und änderbar zu gestalten.

6.3. Das Seitentemplate

In diesem Kapitel wird die Erstellung des Seitentemplates von der HTML-Vorlage bis zur Integration mit Hilfe von TemplaVoila behandelt.

6.3.1. Die HTML-Vorlage

Die grundsätzliche Struktur der Vorlage wird mit Hilfe von DIVs erstellt. Damit später das Layout, welches mit Hilfe von „Cascading Style Sheets“ realisiert wird, auch angezeigt wird, muss bereits in der Vorlage die Zuordnung von Klassen bzw. IDs zu den einzelnen DIVs getroffen werden. Die in Abschnitt 5.1 definierte Gliederung der Webseite findet in der HTML-Vorlage ihre Entsprechung.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
<div id="page">
  <div id="head">
    <div id="topmenu">
      <div id="quicknavi"></div>
      <div id="searchbox"></div>
      <div id="languageselection"></div>
      <div id="tabs"></div>
    </div>
  </div>
  <div id="right">
    <div id="content" style="">
      <div id="breadcrumb"></div>
      <!--TYPO3SEARCH_begin-->
      <div id="innercontent"></div>
      <!--TYPO3SEARCH_end-->
    </div>
  </div>
  <div id="left">
    <div id="menuselect"></div>
    <div id="menu"></div>
    <div id="overflow1"></div>
    <div id="leftcontent"></div>
    <div id="overflow2"></div>
  </div>
</div>
<div id="footer"></div>
</body>
</html>
```

Damit später die Seite XHTML-strict konform sein kann, ist auch die HTML-Vorlage XHTML-strict konform gestaltet. Die Vorlage enthält nur das Gerüst der späteren Seite, sämtliche Inhalte, auch der Seitentitel etc. werden eingefügt.

Die beiden DIVs `overflow1` und `overflow2` enthalten über das CSS-Stylesheet später graphische Elemente. Die beiden Kommentare `<!--TYPO3SEARCH_begin-->` `<!--TYPO3SEARCH_end-->` umschließen den Hauptinhaltsbereich. Dies ist notwendig, um die spätere Ausführung der Extension „indexed_search“ zu optimieren. Diese Extension führt die portalweite Suche durch. Die Kommentare in der Vorlage bewirken eine Beschränkung der Suche auf eben den Hauptinhaltsbereich.

Dieses Template wird unter dem Titel *maintemplate.htm* gespeichert und im Portal unter */fileadmin/templates* zur späteren Verwendung abgespeichert.

6.3.2. Erstellung des Seitentemplates

Nachdem die hochgeladene Datei *maintemplate.htm* in TemplaVoila geladen wurde, muss erst die Data Structure (DS) mit den zu mappenden Feldern erstellt werden. Folgende Felder werden erstellt:

- *quicknavi*
Schnellnavigation. Quicknavi hat den Editing Type „TypoScript Object Path“ (TSOP); die Inhalte können deshalb im Template über `lib.quicknavi` eingefügt werden.
- *languageselection*
Sprachauswahlbutton. Besitzt den Editing Type TSOP und kann im Template über `lib.language` angesprochen werden.
- *tabs*
Hauptmenüpunkte in Listenform. Menüs werden ebenfalls über TypoScript generiert, ansteuerbar über `lib.tabs`.
- *coverimage*
Das Coverimage definiert für jedes Unterportal ein anderes Hintergrundbild hinter dem Hauptinhaltsbereich mit Hilfe des „style“-Attributes. Das Bild wird im Template über `lib.coverimage` eingefügt.
- *content*
Dieses Feld wird später durch die Hauptinhalte ersetzt. Diese werden durch Content Elements im Seiteneditor erstellt, deshalb besitzt `content` den Editing Type „Content Element“.

- *leftmenu*
Enthält später die Untermenüpunkte des aktuellen Hauptmenüpunktes, Editing Type „TypeScript Object Path“, ansteuerbar über lib.leftmenu.
- *footer*
Wird ebenfalls über TypeScript generiert, und wird im Template über lib.footer angesprochen.
- *searchbox*
Ist ebenfalls vom Editing Type TSOP, die Suchbox wird über lib.searchbox eingefügt.
- *menuselect*
Enthält das mit Hilfe einer Selectbox realisierte Schnellwahlmenü, dieses wird im Template über lib.menuselect erstellt.
- *widgetsleft*
Platzhalter für die Security-Widgets, die über lib.widgets im Template eingebunden werden.
- *breadcrumb*
Enthält die Breadcrumb zur Vereinfachung der Navigation, kann im Template unter lib.breadcrumb angesprochen werden.

Das Mapping selbst gestaltet sich einfach, jedes Feld wird durch Klick auf das entsprechende DIV gemappt. Der Mapping Type ist jedes Mal „inner“, damit die Klassen- und ID-Information der Vorlage nicht verloren gehen.

Anschließend wird das Template im SysFolder „Storage Folder“ gespeichert.

6.3.3. Koppelung des Seitentemplates an den Seitenbaum

Um das Template nun auch zur Erstellung von Inhalten benutzen zu können, muss es noch an die Seiten im Seitenbaum gekoppelt werden. Dies geschieht grundsätzlich im Seitentemplate der Hauptseite „Portal“.

Um allerdings die gesamte Konfiguration zu erleichtern, wird ein neues Template im Ordner „Storage Folder“ erstellt, welches in das Template von „Portal“ eingebunden wird. Wie in Abschnitt 4.5.4 erläutert, wird im Template die Main-Funktion von TemplaVoila als Funktion zur Seitenerstellung definiert.

Nun kann auf der Rootseite „Portal“ bei den Seiteneigenschaften im Reiter „Extended“ das Maintemplate als zu benutzende Vorlage eingestellt werden. Diese Einstellung gilt auf Grund der Vererbung für alle Seiten im Seitenbaum.

6.4. Inhalt

Die Erstellung der Inhalte in Typo3 geschieht auf verschiedene Art und Weise. Im Folgenden wird auf die verschiedenen Inhaltstypen auf der Webseite kurz eingegangen.

6.4.1. Die Portalseiten

Wie oben erwähnt, ist der Inhaltsbereich auf den Portalseiten in fünf Sektionen unterteilt. Diese Gliederung wird durch den Einsatz eines „Flexible Content Elements“ erreicht, welches wiederum mit TemplaVoila erstellt wird. Die Struktur der HTML-Datei gliedert sich in fünf DIVs, welche mit Hilfe von CSS formatiert werden.

```
<body>
  <div id="div1">Willkommen</div>
  <div id="div2">Sec News</div>
  <div id="div3">Studiennews</div>
  <div id="div4">Upgeladete Dateien</div>
  <div id="div5">Fall der Woche</div>
</body>
```

Die im Beispiel vorhandenen Inhalte werden beim Mapping durch TemplaVoila gelöscht. DIV1 soll später einen Begrüßungstext bzw. einleitende Worte enthalten, der Mapping-Type in TemplaVoila ist „Rich text editor for bodytext“. Die anderen DIVs werden als „Content Element“ gemappt, sie sollen Neuigkeiten anzeigen bzw. andere Funktionen anbieten, welche durch Plugins realisiert werden. Das Flexible Content Element wird unter dem Namen „Portal Elements“ gespeichert und als Inhalt in die bestehenden Portalseiten eingefügt.

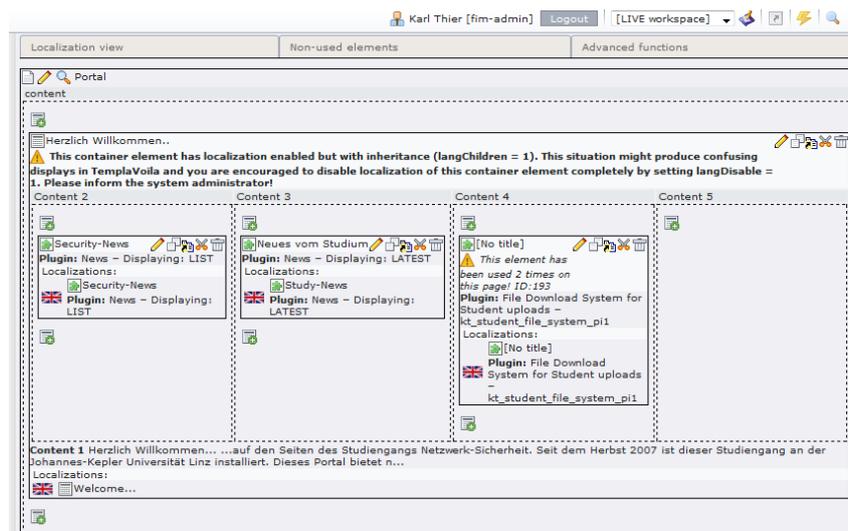


Abbildung 14: Inhalte der Startseite

Wie in der Abbildung der Startseiteninhalte oben zu erkennen, enthält der erste Inhaltsbereich bereits einen Begrüßungstext, während der zweite und dritte Inhaltsbereich jeweils ein Plugin der Extension tt_news enthalten. Im zweiten Inhaltsbereich werden die „Security-News“ als

Liste angezeigt werden (tt_news Typ „List“), im dritten sind die „News zum Studium“ zu sehen (tt_news Typ „Latest“). Im vierten Inhaltsbereich werden die zuletzt hochgeladenen Studentarbeiten vorgestellt. Der fünfte Inhaltsbereich ist zum Start des Portals noch ohne Inhalte.

Genau nach dem gleichen Schema werden auch die anderen Portalseiten mit passenden Inhalten versehen.

6.4.2. Statische Informationen

Die Informationen zum Studium, Übergangsbestimmungen, Voraussetzungen, Impressum, etc. sind in den jeweiligen Seiten mit Hilfe des RichText-Editors realisiert. Dieser bietet neben vielfältigen Möglichkeiten der Textgestaltung und –formatierung auch Funktionen zur Erstellung von Tabellen und Listen.

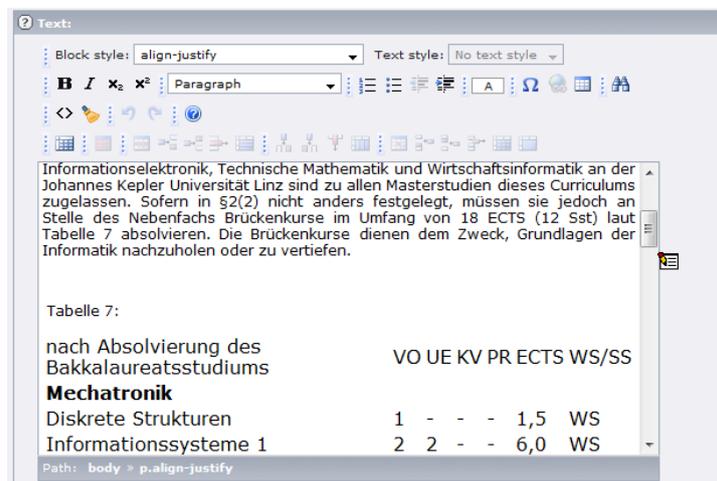


Abbildung 15: RichText Editor mit Teilen des Inhaltes von "Voraussetzungen"

6.4.3. Das Team

Die Präsentation der Personen erfolgt wiederum über ein Flexible Content Element. Dieses enthält neben einem Bild der betreffenden Person und dem Namen auch Kontaktinformationen. Eine Besonderheit dieser Vorlage ist, dass die Kontaktinformationen erst durch Klick auf einen Button neben dem Namen durch JavaScript-Funktionen sichtbar werden. Wie im unteren Ausschnitt des Codes zu sehen, ist der Aufruf der benötigten JavaScript Funktion bereits in der Vorlage inkludiert.

Die verwendete JavaScript-Funktion fungiert als „Slider“ und klappt bei Klick auf den Link das versteckte DIV im konkreten Fall in 750 Millisekunden aus bzw. ändert die Transparenz von 100% auf 0%. Das Einklappen erfolgt in umgekehrter Art und Weise. Dieses JavaScript wird auch an anderen Stellen der Webseite verwendet.

```

<body>
<div class=pers_root>
  <div class="pers_tag"></div>
  <div class="pers_image"></div>
  <div class="pers_tag"></div>
  <div class="pers_name"><span>JRM</span> (<a href="javascript;"
onmousedown="toggleSlide(this.parentNode.nextSibling,750);">mehr...</a
>)</div>
  <div id="persinfoklappbar" style="display:none;overflow:hidden;">
    <div class="pers_tag">Telefon:</div>
    <div class="pers_phone">+43 342555-23</div>
    ...
  </div>
</div>
</body>

```

Eine Besonderheit dieses Flexible Content Elements ist, dass einer Data Structure zwei Template Objects zugeordnet sind. Der Grund liegt in der Mehrsprachigkeit. Obwohl die Struktur des Flexible Content Elements im Deutschen und im Englischen gleich ist, muss die Vorlage sowohl im Deutschen als auch im Englischen existieren. Obiges Beispiel ist der deutschen Vorlage entnommen.

Wie in der Abbildung unten zu sehen, besteht dadurch die Möglichkeit, zwischen deutscher und englischer Vorlage umzuschalten. Die Inhaltsfelder bleiben dadurch gleich, weil ja die gleiche Data Structure verwendet wird, es ändert sich nur die Beschriftung in der Ausgabe. Zu sehen ist ebenfalls, dass es sich bei diesen Inhalten um jene in der „Default“-Sprache handelt, im konkreten Fall ist dies die deutsche Sprache.

In der Abbildung ist auch zu sehen, wie die verschiedenen Editing Types von TemplaVoila zu bearbeiten sind. Das Bild ist als „Image field“ gemappt, Name, Telefon und Raum als „Plain input field“ und Email bzw. „Weitere Informationen“ als „Link field“.

Die Seite „Das Team“ besteht aus einer Folge solcher Flexible Content Elements. Der SysFolder „Personal“ enthält die in die Flexible Content Elements eingebetteten Bilder als Content Element „Image“. Dies ist deshalb erforderlich, weil in den Editing Type „Image field“ von TemplaVoila nur ein Typo3-Image-Objekt eingefügt werden kann, jedoch kein Bild als solches.

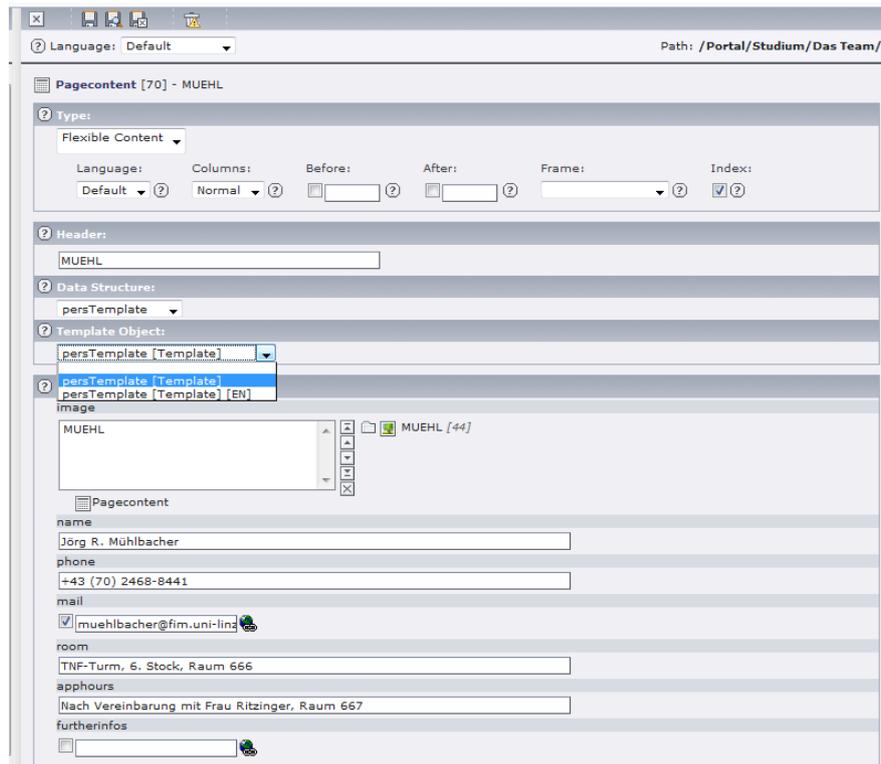


Abbildung 16: Personen-Vorlage

6.4.4. Lehrveranstaltungen

Die Lehrveranstaltungsseiten werden ebenfalls durch ein Flexible Content Element realisiert. Auch hier sind einer Data Structure zwei Template Objects in Form einer deutschen sowie einer englischen Vorlage zugeordnet. Eine Besonderheit ist wiederum die Einbindung des oben erwähnten JavaScripts, das die Upload-Möglichkeit für Studenten ausklappbar macht.

Die Felder Semester, Name, LVA-Nr., LVA-Typ, Wochenstunden und ECTS-Punkte werden als Plain input field gemappt, da sie ja nur einzelne Wörter enthalten.

Inhalt, Literatur und Prüfungsmodus werden als „Rich text editor for bodytext“ gemappt, da diese Felder einen formatierbaren Text enthalten sollen.

Anmeldung wird als „Link field“ angelegt und soll später auf das KUSSS-System verweisen.

Die Felder Dateiupload und Dateien werden später die Funktionen für Dateiup- und -download von Studentendarbeiten enthalten. Das Feld Termine wird durch eine automatische Funktion die aus dem KUSSS-System extrahierten Termine anzeigen. All diese Funktionen werden durch Extensions realisiert, welche in Form von Frontend-Plugins in die Seite eingebunden werden. Deswegen werden diese Felder als TypoScript Object Path gemappt. Das Feld Vortragender wird als Content Element gemappt. Um Redundanz zu vermeiden, werden die einzelnen Vortragenden im Seiteneditor als Content Element vom Typ „Insert

record“ eingefügt. Mit diesem Typ kann man beliebige andere Elemente im Seitenbaum wiederverwenden. Hier wird einfach auf die Personen-Einträge, welche auf der Seite „Das Team“ bereits existieren, verwiesen. So muss eine eventuelle Änderung nur dort vorgenommen werden und ist trotzdem auch auf den Lehrveranstaltungsseiten sichtbar.

Die Lehrveranstaltungen sind jeweils einem thematischen Portal zugeordnet. Trotzdem soll das Unterportal Studium natürlich eine Liste aller Lehrveranstaltungen enthalten. Diese Liste wird aus eigenen Seiten vom Typ „Shortcut“ aufgebaut, welche auf die eigentlichen LVA-Seiten in den Unterportalen verweisen.

6.5. Erstellung von Menüs

Die Navigation auf *security.fim.uni-linz.ac.at* basiert auf verschiedenen, dynamisch generierten Menüs. Die Menükonfiguration ist im Template „Menü“ im SysFolder „Storage Folder“ gespeichert. Wie diese Menüs mit Hilfe von TypoScript erstellt werden, wird im folgenden Abschnitt erläutert.

6.5.1. Das Hauptmenü

Wie in der Abbildung unten zu sehen, ist das Hauptmenü dreistufig. Bei Berührung der immer sichtbaren Hauptmenüpunkte mit der Maus, werden die Untermenüpunkte nach unten ausgeklappt. Eine optionale dritte Ebene an Menüpunkten öffnet sich bei Berührung des Untermenüpunktes nach rechts sowie leicht nach unten versetzt.



Abbildung 17: Das Hauptmenü

Die Struktur des Hauptmenüs wird durch verschachtelte Listen gebildet, die mit Hilfe von CSS formatiert und in die obige Form gebracht werden.

Das Hauptmenü wurde in der TemplaVoila-Vorlage mit dem Editing Type „TypoScript Object Path“ angelegt, kann also mit TypoScript in einem Template konfiguriert werden. Der zugehörige Pfad ist `lib.tabs`.

In der ersten Zeile des Sourcecodes wird `lib.tabs` ein hierarchisches Menü (HMENU) zugewiesen. Der Zugriff auf die verschiedenen Ebenen des Menüs erfolgt nun über `lib.tabs.1` bis `lib.tabs.3`. (Ebene 0 wäre die Wurzel des Seitenbaumes, „Portal“).

Die erste Ebene des Menüs soll aus einem TMENU, aus einem textuellen Menü bestehen. Das Menü soll ja in Listenform ausgegeben werden, weswegen das gesamte Menü mit den Tags für nicht sortierte Listen `` umschlossen wird.

`expAll=1` bewirkt, dass nicht nur die Untermenüpunkte des aktuell ausgewählten Menüs angezeigt werden, sondern alle Untermenüpunkte.

Die einzelnen Menüeinträge können in TypoScript durch ihren aktuellen Status angesprochen werden. Alle derzeit nicht gewählten Menüeinträge werden mit `NO` angesprochen, der Ausgewählte mit `CUR.ACT` wählt die „Rootline“, also den aktuellen Eintrag und seine „Vatereinträge“.

Jeder einzelne Menüeintrag soll ein Listeneintrag sein, also mit `` umschlossen werden. `stdWrap.htmlSpecialChars` bewirkt, dass eventuelle Sonderzeichen in ihre HTML-Codes umgewandelt werden.

```
lib.tabs = HMENU
lib.tabs.1 = TMENU
lib.tabs.1 {
    wrap = <ul>|</ul>
    expAll = 1
    NO{
        wrapItemAndSub = <li>|</li>
        stdWrap.htmlSpecialChars = 1
    }
    ACT < .NO
    ACT = 1
    ACT{
        ATagParams = class="act"
    }
}
lib.tabs.2 < lib.tabs.1
lib.tabs.3 < lib.tabs.1
```

Mit der Zeile `ACT < .NO` werden die Einstellungen für `NO` auf `ACT` kopiert, zusätzlich wird für die aktuell gewählten Menüpunkte ein Attribut `class="act"` gesetzt, um diese speziell formatieren zu können. Abschließend werden alle Einstellungen, die für die erste Ebene gelten, auch auf die zweite und dritte Ebene kopiert.

6.5.2. Rootline

Die Rootline soll dem Benutzer Auskunft darüber geben, wo er sich innerhalb der Webseite thematisch und strukturell befindet.



Abbildung 18: Rootline

Auch die Breadcrumb wird über TypoScript in die TemplaVoila-Vorlage eingefügt.

Breadcrumb ist ebenfalls ein hierarchisches Menü, allerdings vom Spezial-Typ „Rootline“.

Range gibt an, welche Ebenen in der Rootline angezeigt werden sollen. Begonnen werden soll bei der Ebene 0, also bei „Portal“. Die Angabe von -1 bewirkt, dass die Rootline bis zum aktuellen Element ausgegeben wird. Die Rootline wird mit Text befüllt, vor jedem Eintrag soll „>“ angezeigt und die einzelnen Einträge werden fett gedruckt.

```
lib.breadcrumb = HMENU
lib.breadcrumb{
    special = rootline
    special.range = 0|-1
    1 = TMENU
    1.NO.allWrap = >&nbsp;|</b>&nbsp;
}
```

6.5.3. Schnellwahlmenü

Das Schnellwahlmenü soll dem Benutzer den schnellen Wechsel zu jeder beliebigen Seite ermöglichen.

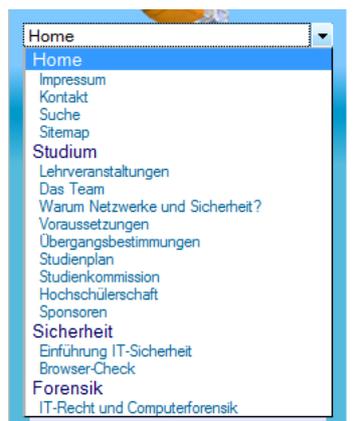


Abbildung 19: Schnellauswahlmenü

Mit Typo3 kann diese Art von Menü auf zwei Arten realisiert werden. Entweder als „TMENU“ oder als „JSMENU“. Aus Gründen der Barrierefreiheit sowie der besseren Durchsuchbarkeit von Suchmaschinen wird dieses Menü als TMENU realisiert.

Eine Besonderheit dieses Menüs ist die Einbettung in eine Selectbox.

```
wrap = <div><select class="dropdownmenu" name="dropdown_navi" size="1"
onChange="document.location.href=this.options[this.options.selectedIndex].value" > | </select></div>
```

Weiters gilt, dass die Einträge in diesem Menü nicht selbst als Hyperlinks auf die neue Seite verlinken – diese Aufgabe übernimmt die onChange-Methode der Select-Box. In TypoScript werden die einzelnen Einträge durch die Zeile `doNotLinkIt = 1` ohne Hyperlink gerendert.

Der Pfad wird stattdessen in den `value`-Wert des Optionsfeldes geschrieben. Dazu wird ein „typolink“ auf die Seiten-ID des aktuellen Menüeintrages mit `field:uid`, erstellt. Ein typolink enthält normalerweise einen HTML-Link auf `parameter.data` – durch die Angabe von `returnLast = url` wird allerdings die URL ohne Link gespeichert. Diese URL wird durch `wrap` in den `option`-Tag des Optionsfeldes geschrieben. Um die absolute Web-Adresse der zu verlinkenden Seite zu erzeugen, wird die Domain durch Aufruf von `{ $BASE_URL }` eingefügt. `{BASE_URL}` ist eine im Haupttemplate definierte Konstante.

Mit der Funktion `before` wird der gesamte Tag vor dem eigentlichen Menütex eingebunden und am Ende der schließende `</option>`-Tag eingefügt.

```
1 {
  expAll = 1
  NO {
    doNotLinkIt=1
    before{
      typolink.parameter.data = field:uid
      typolink.returnLast = url
      wrap = <option class="menulevell1" value="{ $BASE_URL }| ">
    }
    allWrap = | </option>
  }
}
```

Genauso wird mit dem CUR-Menüeintrag verfahren, wobei hier in den Option-Tag zusätzlich noch das `selected="selected"` Attribut eingefügt wird, damit die aktuell ausgewählte Seite im Auswahlfeld auch an der ersten Stelle angezeigt wird.

6.5.4. Seitliches Menü

Da die Untermenüpunkte des aktuell gewählten Hauptmenüeintrages durch die Positionierung des Hauptmenüs nicht auch dort permanent angezeigt werden können, werden sie in ein separates Menü am linken Seitenrand ausgelagert.

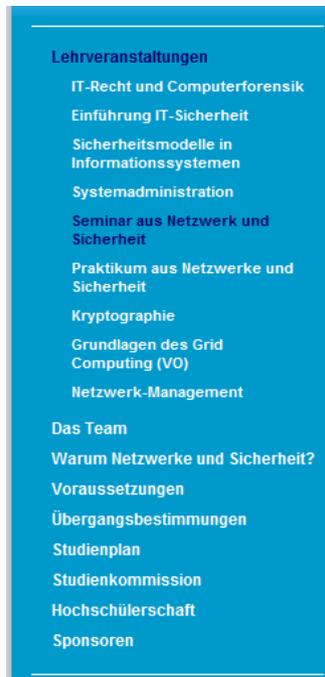


Abbildung 20: Menü am linken Seitenrand

Prinzipiell ist dieses Menü genau gleich aufgebaut wie das Hauptmenü. Es wird genauso eine Liste ausgegeben, welche mit Hilfe von CSS entsprechend formatiert wird. Der einzige Unterschied ist, dass `entryLevel = 1` gesetzt wurde – denn die Hauptmenüpunkte sollen in diesem Menü nicht angezeigt werden.

6.6. Extensions

Im Folgenden werden die verwendeten Extensions vorgestellt sowie ihre Konfiguration und Verwendungsweise erläutert.

6.6.1. tt_news

Das Newssystem von *security.fim.uni-linz.ac.at* inkludiert redaktionell erstellte Nachrichten für studienbezogene Neuigkeiten und automatisch aggregierte Nachrichten aus diversen externen Quellen aus dem Internet.

Diese Funktionalität wird, bis auf die Aggregation der externen Nachrichten, von „tt_news“ erledigt. Diese bietet unter anderem folgende Features:

- Kategorisierbare Newseinträge

- Drei verschiedene Newstypen:
 - Klassische Nachrichten mit Header, Vorschautext, Text (für Studiennachrichten notwendig)
 - Link zu einer internen Seite
 - Link zu externen Seiten (dieser Typ wird für die aggregierten Nachrichten aus den externen Quellen verwendet, aus rechtlichen Gründen müssen extern eingebundene Feeds auf den externen Artikel verweisen)
- Automatische Archivierung nach einer definierten Zeitspanne
- Verschiedene, konfigurier- und anpassbare Modi der Ansicht von News (Liste aller Newseinträge („List“), Liste der letzten Newseinträge („Latest“), einzelner Newseintrag in Vollansicht („Single“), Archiv, Suche)

6.6.1.1. Konfiguration

Um die Konfiguration von tt_news kompakt zu halten, wird ein eigenes Template „News“ für die Erweiterung im „Storage Folder“ erstellt. Dieses Template wird in das Template „Root“ eingebunden und ist dadurch auf dem ganzen Portal gültig.

Newseinträge können in tt_news entweder nach einer gewissen Zeitspanne automatisch oder zu einem gewissen, vom Benutzer eingestellten Zeitpunkt archiviert werden. Diese Archivierungsmöglichkeiten schließen sich jedoch gegenseitig aus.

Folgende Zeilen aktivieren die Archivfunktion, deaktivieren die manuelle Einstellung des Archivierungsdatums und setzen die Frist zur Archivierung auf zehn Tage fest. Weiters wird die ID jener Seite angegeben, welche das Archiv enthält.

```
plugin.tt_news {
  archive=1
  enableArchiveDate = 0
  datetimeDaysToArchive = 10
  archiveTypoLink.parameter=54
```

tt_news erlaubt vielfältige Konfigurationsmöglichkeiten der Datumsausgabe.

```
archiveTitleCObject {
  10.strftime = %B - %Y
}
getRelatedCObject {
  20.strftime = %d.%m.%Y %H:%M
}
displayList {
  date_stdWrap.strftime= %d. %m. %Y
  time_stdWrap.strftime= %d.%m.%y %H:%M
  age_stdWrap.age = &nbsp;Min.| Std.| Tage| Jahre
} }
```

Die Ausgabe muss für die englische Sprache angepasst werden. Mit einer Bedingung, einer „Condition“ wird die aktuell ausgewählte Sprache überprüft.

```
[globalVar = GP:L = 1]
plugin.tt_news.displayList {
    age_stdWrap.age = &nbsp;minutes| hours| days| years
}
[global]
```

Für die Speicherung der Nachrichten und der Newsgroups wird ein SysFolder „News“ im Seitenbaum erstellt.

6.6.1.2. Newskategorien

Um Nachrichten gruppieren zu können, werden Newskategorien benötigt. Diese werden im SysFolder „News“ erstellt. Dazu öffnet man den Ordner im „List-View“ – Modus und fügt eine „News category“ hinzu. Eine News category besteht aus folgenden Feldern:

- *Title*
Ist der Name der Kategorie.
- *Title language overlay*
Enthält den Namen der Kategorie in anderen Sprachen.
- *Parent category*
Kategorien können auch geschachtelt werden. Im Feld „parent category“ kann die übergeordnete Kategorie ausgewählt werden.
- *Category image*
Eine Kategorie kann neben dem Namen auch durch ein Bild repräsentiert werden, welches neben einem Newseintrag angezeigt werden kann. Im Security-Portal werden die Logos der Quellseiten (z.B. www.heise.de) als „category images“ verwendet.
- *News category shortcut*
Bei Klick auf das category image bzw. auf den title einer Kategorie kann auf eine interne oder externe Seite verlinkt werden.
- *Single-view page for news from this category*
Wird auf einer Seite ein in einer Liste befindlicher Newseintrag ausgewählt, so wird dieser auf einer eigenen Seite angezeigt. Diese Seite kann für jede Kategorie extra definiert werden.

Für *security.fim.uni-linz.ac.at* wird eine fixe Kategorie „Studium“ definiert, es ist dies jene Kategorie, der die Neuigkeiten das Studium betreffend zugeordnet sein werden.

Weitere Kategorien werden in Abhängigkeit von den gewählten Nachrichtenquellen angelegt, sodass jede Quelle einer Kategorie entspricht.

6.6.1.3. Newseinträge

Auch Newseinträge werden als Element in einem SysFolder erstellt und gespeichert. In Abhängigkeit vom gewählten Typ stehen für einen Newseintrag verschiedene Eigenschaften zur Verfügung.

- *Title*
Ist der Titel des Newseintrages und gleichzeitig seine Überschrift.
- *Date/Time*
Hier kann das Erstelldatum konfiguriert werden; als Voreinstellung ist der aktuelle Zeitstempel eingetragen.
Weiters kann ein Archivierungsdatum, sowie die Sprache des aktuellen Newsbeitrages gesetzt werden.
- *Author*
Name und Email-Adresse des Autors können eingetragen werden.
- *Subheader*
Der Subheader enthält den Teaser-Text des Newsbeitrages, also eine kurze Zusammenfassung des Artikels bzw. seinen ersten Absatz. Dieser wird bei Darstellung des Newsbeitrages in Listenform mit dem Titel angezeigt und soll einen Kurzüberblick über den Artikel bieten.
- *Text (nur Typ „News“)*
Enthält den eigentlichen Text des Newseintrages. Dieser ist mit dem Typo3-eigenen Rich-Text-Editor voll formatierbar. Der Text wird nur im Single-Mode angezeigt.
- *Shortcut to page (nur Typ „Link internal page“)*
Hier kann eine Seite des Seitenbaums ausgewählt werden, auf die dieser Newseintrag verweisen soll. „Link internal page“-Nachrichten können nicht im Single-Mode angezeigt werden, bei Klick auf die Überschrift verweisen sie direkt auf die interne Seite.
- *External URL (nur Typ „Link external URL“)*
Enthält die externe URL, auf die der Newseintrag verweisen soll. Auch „Link external URL“-Newseinträge können nicht im Single-Mode angezeigt werden.
- *Keywords*
Für jeden Newseintrag können bestimmte Keywords vergeben werden, welche im Header der Seite als Metatags eingefügt werden. Dies sorgt unter anderem für leichtere und effizientere Indizierung durch Suchmaschinen.

- *Category*
Jeder Newsbeitrag kann beliebig vielen Kategorien zugeordnet sein.
- *Images + Caption*
In einem Newsbeitrag können auch Bilder vorkommen. Zu den Bildern können auch noch eine Beschriftung, sowie „alt-“ und „title-Tag“ verfasst werden.
- *Links*
Zu jedem Newsbeitrag kann eine beliebige Anzahl an Links definiert werden; diese werden unter dem Text im Single-Mode angezeigt.
- *Related news*
Existieren zu einem Newsbeitrag in Zusammenhang stehende News oder Seiten, so können diese ebenfalls verlinkt werden.
- *Files*
Auch eine beliebige Anzahl an Dateien kann mit einem Newsbeitrag verknüpft werden.

6.6.1.4. News-Plugin

tt_news wird in eine Seite durch ein Plugin integriert. Dieses kann im Seiteneditor als einfaches Content Element eingebunden werden und erlaubt folgende Einstellungen:

General settings

- *What to display*
Diese Eigenschaft bestimmt, wie die gewählten Nachrichten angezeigt werden.
„List“ zeigt alle Nachrichten der gewünschten Kategorie als sequentielle Liste.
„Latest“ zeigt eine gewisse Anzahl an Nachrichten der Kategorien.
„Single“ zeigt genau einen Newseintrag, mit allen zugehörigen Informationen.
„Amenu“ erlaubt eine Navigation durch das Archiv auf Basis des Erstelldatums eines Beitrags. Die Auswahl erfolgt nach Monaten und Jahr, sortiert von jetzt zurück.
„Search“ rendert ein Suchfeld, mit dem die Newseinträge durchsucht werden können.
„Catmenu“ erlaubt die Navigation mit Hilfe der Kategorien.
„Version_Preview“ sorgt dafür, dass News-Beiträge vor der Freischaltung über einen speziellen Button im Erstellfenster vorab betrachtet werden können.
Es können beliebig viele dieser Anzeigemodi in einer Instanz des Plugins zusammengefasst werden. Dies ist insbesondere sinnvoll, wenn eine Navigationskomponente mit einer Präsentationskomponente gemeinsam eingesetzt wird, z.B. Amenu mit dem List-View.

- *Order by*
Die Eigenschaft „Order by“ ist nur in Kombination mit List und Latest sinnvoll und erlaubt die Sortierung der Einträge nach Zeitstempel, Archivierungsdatum, Autor, Titel, Typ oder nach Zufallsmuster.
- *Ascending or Descending*
Gibt an, in welche Richtung die Sortierung erfolgen soll.
- *Category mode/Category selection*
Mit der „Category selection“ können beliebige Kategorien ausgewählt werden. Der „Category mode“ bestimmt, was mit den ausgewählten Kategorien passieren soll.
„Show all“ zeigt alle Einträge, unabhängig von der Auswahl in der Category selection.
„Show items with selected Category (or)“ zeigt alle Einträge, die mindestens einer der ausgewählten Kategorien zugeordnet sind.
„Show items with selected Category (and)“ zeigt alle Einträge, die in jeder der ausgewählten Kategorien eingetragen sind.
„Do NOT show items [...] (or)“ zeigt alle Einträge die NICHT mindestens einer der Kategorien zugeordnet sind.
„Do NOT show items [...] (and)“ wiederum zeigt jene Einträge, welche NICHT in allen der gewählten Kategorien eingetragen sind.
- *Use subcategories*
Gibt an, ob Subkategorien ebenfalls in die Auswahl einbezogen werden sollen. Mögliche Werte sind „Yes“, „No“ und „Defined by TypoScript“.
- *Archive setting (nur für Typ List relevant)*
Gibt an, ob archivierte, nicht archivierte oder alle Newsbeiträge in der Liste angezeigt werden.
- *Startingpoint/Recursive*
Hier muss der SysFolder bzw. die Seite angegeben werden, welche die Newsgruppen und -einträge enthält.
Die Einstellung „Recursive“ gibt an, ob auch Newsgruppen und -einträge von Unterordnern bzw. Unterseiten des Startingpoints miteinbezogen werden sollen.

Template

- *Template file reference*
Das Design der einzelnen Typen der Newsrepräsentation (List, Latest,...) kann vollständig benutzerdefiniert werden. Dies geschieht über HTML-Templates mit Markern. Die Templates werden von tt_news geparkt und dynamisch mit Inhalten

gefüllt. Jede Instanz des Plugins kann ein anderes Template benutzen, was eine sehr flexible Formatierung der Nachrichtenpräsentation ermöglicht. In der „Template file reference“ wird der Pfad zum zu benutzenden Template festgelegt. Als Beispiel wird hier das Template für die „Latest“-Ansicht der Studiums-Nachrichten dargestellt. In diesem erkennt man die Marker, welche von Typo3 dynamisch durch Inhalte ersetzt werden. Der umgebende HTML-Code ist mit DIVs und CSS-Klassen gestaltet, mit denen die Struktur bzw. das Layout der News festgelegt wird. Wie viele Newselemente schlussendlich im Container eingefügt werden, hängt von den Einstellungen des Plugins bzw. des Typo3-Templates ab.

```

<!-- ###TEMPLATE_LATEST### begin
    This is the template for the latest news, typically displayed on a
    frontpage -->
    <div class="news-latest-container">
<!-- ###CONTENT### begin
This is the part of the template substituted with the list of news:
-->
<!-- ###NEWS### begin Template for a single item -->

    <div class="news-latest-item">
        <h3>
            <!--###LINK_ITEM###-->###NEWS_TITLE###<!--
            ###LINK_ITEM###-->
        </h3>
        <!--###LINK_ITEM###-->###NEWS_IMAGE###<!--###LINK_ITEM###-->

        <div class="news-latest-date">
            ###NEWS_DATE### ###NEWS_TIME###
        </div>
        ###NEWS_SUBHEADER###
        <div class="clearer"></div>
        <div class="news-latest-morelink">
            <!--###LINK_ITEM###-->###MORE###<!--###LINK_ITEM###-->
        </div>
    </div>
<!-- ###NEWS### end-->
<!-- ###CONTENT### end -->

<div class="news-latest-gotoarchive">
    <!--###LINK_ARCHIVE###-->###GOTOARCHIVE###<!--
    ###LINK_ARCHIVE###-->
</div>
</div>
<!-- ###TEMPLATE_LATEST### end -->

```

- *Weitere Einstellungen*

Im Reiter „Template“ lassen sich noch weitere Einstellungen bezüglich maximaler Höhe und Breite von Bildern in Newsbeiträgen, sowie diverse Templateeinstellungen festlegen.

Category Settings

- *Category Image Link Mode/Category Text Link Mode*

Mit dieser Einstellung kann festgelegt werden, ob und wie Kategorietitel bzw. -bild verlinkt werden sollen. Möglich sind die Einstellungen „gar nicht anzeigen“, „ohne Verlinkung anzeigen“, „zu Kategorieauswahl verlinken“, „zu voreingestelltem Ziel verlinken“ bzw. „TypoScript-Einstellung verwenden“.

- *Max width/height/number of category images/texts*

Gibt Formatierungsoptionen für den Kategorietitel und das -bild an.

Other Settings

- *“Page-ID for single news display” bzw. “Page-ID to return to” (Einstellung nur für List/Latest sinnvoll)*

Gibt an, auf welcher Seite sich die Instanz des Plugins befindet, in der ein einzelner Newseintrag in der Vollansicht angezeigt werden soll, bzw. auf welche Seite der „Zurück“-Link verweisen soll.

- *“Don’t display first image in single view” bzw. “Don’t display first image in single view even when only one image is available”*

Gibt an, ob das erste dem Newsbeitrag zugeordnete Bild nur in der Vorschau angezeigt werden soll oder nicht.

- *„List StartID“ bzw. „-Limit“*

Diese Einstellungen erlauben die Angabe einer Start-ID, ab welcher die Newseinträge angezeigt werden sollen, weiters die Festlegung der maximalen Einträge in List bzw. Latest.

- *„Don’t display Pagebrowser in List“*

In der Einstellung „List“ wird ein Pagebrowser gerendert, welcher den Zugriff auf ältere Nachrichten ermöglicht. Der Browser kann auch versteckt werden.

- *“Insert pagebreak in SINGLE after this number of words”*

Zusätzlich kann nach einer gewissen Anzahl von Wörtern im Single-Modus der Artikel geteilt werden, um ein leichteres Lesen zu ermöglichen. Am unteren Rand befindet sich dann ein Hyperlink zum nächsten Teil des Artikels.

6.6.2. xml_ttnews_import

Ein wesentliches Feature von *security.fim.uni-linz.ac.at* ist die Aggregation, Gruppierung und Präsentation von aktuellen Nachrichten aus Fremdquellen. Die Gruppierung und Präsentation sollen dabei durch die Extension *tt_news* realisiert werden. *tt_news* bietet jedoch keine Funktionalität für die Aggregation aus XML-Quellen. Diese muss durch eine zusätzliche Extension realisiert werden, „*xml_ttnews_import*“.

6.6.2.1. Installation und Konfiguration

Bei der Installation wird automatisch auch die Extension „*cc_rdf_news_import*“ mitinstalliert, von der *xml_ttnews_import* abhängig ist. Die Extension ermöglicht die Erstellung von „News feed imports“. Wie in Abbildung 21 erkennbar, bietet ein „News feed import“ zahlreiche Einstellungsmöglichkeiten an, von denen die meisten von der Basisextension herrühren und in der aktuellen Extension keine Bedeutung mehr haben.



Abbildung 21: Erstellung von News feed imports

Wichtig sind folgende Einstellungsmöglichkeiten:

- *RDF-URL*

Hier wird die URL des zu importierenden RSS-Feeds angegeben

- *Target page for tt_news import*

In diesem Feld muss jener SysFolder ausgewählt werden, in den die Newseinträge gespeichert werden sollen. Für das Security-Portal ist dies der Ordner „News“.

Die Aktualisierung der Nachrichten muss händisch erfolgen, durch einen Mausklick im Backend auf den jeweiligen news feed import.

Dies ist für das Sicherheitsportal nicht ausreichend, da die Aktualisierung aus allen Nachrichtenquellen automatisiert und zeitgesteuert erfolgen soll. Weiters bietet die

Erweiterung keine Unterstützung für Atom-Feeds und erlaubt keine Zuordnung von Nachrichten oder Nachrichtenquellen zu tt_news-Gruppen.

6.6.2.2. Änderung der Extension mit Hilfe des Kickstarters

Die Zuordnung zu Gruppen soll ähnlich wie die Zuordnung zu einer Seite bzw. einem SysFolder bei der Erstellung eines News feed import im Backend erfolgen. Um dieses zusätzliche Feld, das einem Tabellenfeld entspricht, hinzuzufügen, ist der „Kickstarter“ ein geeignetes Werkzeug. Dieser ermöglicht auch die Bearbeitung bereits installierter Extensions, überschreibt aber jeglichen Programmcode beim Export. Der Code muss daher vorher gesichert und nachträglich wieder eingefügt werden.

The screenshot shows the 'Extension Manager' interface for the extension 'Import an RSS feed into tt_news (xml_ttnews_import_extended)'. The 'KICKSTARTER WIZARD' is active, and the 'Extend existing Tables' section is selected. The wizard is configuring a new field named 'group' for the 'News feed import (tx_ccrdfnewsimport)' extension. The field is being added to the 'Pages table, (pages)'. The field type is 'Database relation', and it is marked as an 'Exclude-field'. The field title is 'tt_news-Group for RSS-Feed'. The wizard also shows options for creating a relation to a table, with the 'Pages table, (pages)' selected. The field type is 'Field with Element Browser', and the 'Max number of relations' and 'Size of selector box' are both set to 1. The 'True M-M relations (otherwise commalist of values)' checkbox is unchecked. The wizard also shows options for creating a relation to a table, with 'Custom table (enter name below)' selected and the 'Custom table name' set to 'tt_news_cat'. The field type is 'Field with Element Browser', and the 'Max number of relations' and 'Size of selector box' are both set to 1. The 'True M-M relations (otherwise commalist of values)' checkbox is unchecked.

Abbildung 22: Bearbeitung einer Extension im Kickstarter

In Abbildung 22 ist das neue Feld bereits hinzugefügt. Die wählbaren Elemente stammen aus der Datenbank-Tabelle „tt_news_cat“. Der Extension-Key wurde ebenfalls angepasst, auf „xml_ttnews_import_extended“, um den Unterschied zur Stammextension zu dokumentieren. Mit Klick auf den Button „View result“ und dem anschließenden Klick auf „Write“ kann die Extension direkt installiert werden.

Um Atom-Feeds lesen zu können und die Extension automatisiert ausführbar zu machen, sind Änderungen im Quellcode unerlässlich.

- *Lesen von Atom-Feeds*

Die Extension mappt die Felder von RSS in Datenbankfelder von tt_news. Dies passiert, indem Anfangs- und Endtag eines Eintrags erkannt werden und der Inhalt des Tags in die dazugehörige Variable (und damit in das entsprechende Tabellenfeld von tt_news) eingetragen wird.

Exemplarisch wird dies hier für den Begin-Tag von Channel, Title und Description in einem RSS-Feed gezeigt:

```
if ($name == "CHANNEL") {  
    $blChannel = True;  
} elseif ($name == "TITLE") {  
    $blTitle = True;  
} elseif ($name == "DESCRIPTION") {  
    $blDescription = True;  
}
```

Damit die Erweiterung Atom-Feeds lesen kann, müssen einfach die entsprechenden Tags, die Atom-Feeds enthalten, auf die richtigen Variablen gemappt werden.

Im konkreten Fall sind dies folgende vier Tags:

```
ENTRY -> $blItem  
ID -> $blLink  
SUMMARY ->$blDescription  
UPDATED -> $blPubDate
```

Da im Newssystem ohnehin nur diese Informationen vonnöten sind, wurde auf das Mapping von zusätzlichen Feldern verzichtet.

- *Automatisierbare Ausführung*

Die Aktualisierung der Nachrichten soll mit Hilfe des Windows Task Managers automatisiert erfolgen. Dieser soll zu jeder vollen Stunde ein PHP-Skript aufrufen.

Dieses Skript ist in der Extension für die Möglichkeit des manuellen Updates bereits enthalten. Um dieses Skript auch von externer Stelle aufrufbar zu machen, ist eine Säuberung des Codes nötig. Der gesamte Typo3-interne Code bzw. die Ausgabe wird gelöscht, ebenso die Klassendefinition. So kann von externer Stelle die Methode updateRSS() aufgerufen werden, welche das Update durchführt. Einziger Nachteil ist, dass eine externe Datenbankverbindung hergestellt werden muss, da ja die Typo3-internen Datenbankfunktionen nicht zur Verfügung stehen.

6.6.3. „realurl“

Typo3 basiert auf PHP, das heißt, prinzipiell wird die Frontend-Ausgabe durch das PHP-Skript *index.php* generiert, welchem die Seiten-ID der zu rendernden Seite übergeben werden muss. Die daraus resultierenden URLs sind von der Form *<Domain>/index.php?id=34*. Besser, sowohl für Besucher als auch für Suchmaschinen-Bots, ist die Verwendung von sprechenden URLs, etwa von der Form *<Domain>/studium/voraussetzungen.html*.

Die Extension „realurl“ ermöglicht genau diese Form der URLs und ist dabei für den Benutzer wie auch den Entwickler vollständig transparent. Links sowie alle Menüeinträge verweisen automatisch auf die sprechenden URLs von Typo3. Wird eine bestimmte Seite über die sprechende URL aufgerufen, so werden aus dieser URL automatisch eventuelle GET/POST-Parameter extrahiert und daraufhin der Renderingprozess ganz normal mit der *index.php* gestartet. Bei der Ausgabe wird dieser Prozess genau umgekehrt. Die Parameter der *index.php* werden in die sprechende URL übersetzt.

6.6.3.1. Konfiguration

Realurl erzeugt URLs, welche auf dem verwendeten Server jedoch nicht existieren. Der Server muss daher ebenfalls für die Verwendung von realurl konfiguriert werden, um die http-Fehlermeldung 404, „Not Found“ zu vermeiden.

Die Regeln, wie realurl die URLs encodiert bzw. decodiert, werden in einer PHP-Datei als PHP-Array gespeichert.

Realurl ermöglicht die automatische Konfiguration, welche die installierten Extensions und Sprachen berücksichtigt. Für eigene Anwendungen können die Mapping-Regeln aber jederzeit verändert werden.

Die Einbindung bzw. Aktivierung von realurl funktioniert über TypoScript mit folgenden vier Zeilen:

```
config.simulateStaticDocuments = 0
config.baseURL = {$BASE_URL}
config.tx_realurl_enable = 1
config.prefixLocalAnchors = all
```

`simulateStaticDocuments` ist ebenfalls eine Extension, führt ähnliche Funktionen wie realurl aus und muss deshalb deaktiviert werden. Die `baseURL` ist der Domainname, in diesem Falle also *security.fim.uni-linz.ac.at*. Die dritte Zeile aktiviert realurl. `prefixLocalAnchors=all` sorgt dafür, dass alle Links inklusive BaseURL als absolute Links gerendert werden.

6.6.4. indexed_search + crawler

Ein Webportal mit vielen, vor allem auch textuell sehr umfangreichen Inhalten benötigt eine ausgereifte Suchfunktion. Diese soll nicht nur die Suche nach den Inhalten der einzelnen Seiten, sondern auch die Suche in den hochgeladenen Dokumenten ermöglichen.

Die Systemerweiterung „indexed_search“, erweitert mit der Extension „crawler“, erfüllt genau diese Anforderungen.

6.6.4.1. indexed_search

Die Erweiterung indexed_search ist eine Eigenentwicklung von Kasper Skårhøj und wird inzwischen als System-Extension mit jeder Installation von TYPO3 mitgeliefert, ist jedoch im Installationszustand nicht geladen.

Zur Einbindung der Suchfunktion in die Seite bietet indexed_search ein Frontend-Plugin. Das Plugin fungiert als Suchformular, welches über TypoScript in einem Template konfiguriert werden kann. Grundsätzlich bietet das Formular sowohl eine einfache als auch eine erweiterte Suchfunktion an, wobei für letztere zahlreiche zusätzliche Einstellungsmöglichkeiten im Template vorhanden sind.

Auch die Suchergebnisse werden im Plugin angezeigt und dort nach Relevanz sortiert. Die Relevanz jedes Suchergebnisses wird auch mit einer kurzen Vorschau untermauert, zusätzliche Informationen sind Größe der gefundenen Datei bzw. Seite, Erstellungs- und Änderungsdatum. Auch der relative Pfad zur Hauptseite wird angezeigt.

Die Indizierungsfunktion von indexed_search wird durch den Aufruf einer Seite gestartet. Dies hat den Nachteil, dass eine nie aufgerufene Seite auch nicht durchsucht wird. Außerdem wird eine Seite genau zu dem Zeitpunkt indiziert, an dem sie auch abgerufen wird, was zu Einbußen in der Performance führen kann.

Zusätzlich gilt, dass nur jene Seiten indiziert werden, die im Cache liegen. Ist in den Seiteneigenschaften einer Seite die Einstellung „no cache“ gewählt, so wird diese ebenfalls nicht indiziert und damit auch nicht durchsucht.

Indexed_search ermöglicht auch die Indizierung von bzw. die Suche in externen Dateien. Dazu benötigt indexed_search Tools wie „catdoc“ für Word-Dokumente oder „pdf2text“ für PDF-Dateien. Diese sind nicht in indexed_search integriert, müssen also nachträglich auf dem verwendeten Server installiert werden. Die Pfade zu den Tools müssen im Extension Manager bei der Konfiguration von indexed_search eingetragen werden.

Es werden nur jene Dateien durch indexed_search indiziert, welche auf den indizierten Seiten direkt verlinkt werden.

6.6.4.2. crawler

crawler ist eine weitere Extension von Kasper Skårhøj und wird meist in Kombination mit indexed_search verwendet.

Prinzipiell ermöglicht crawler händische bzw. durch einen Cron-Job angestoßene Operationen auf den Seiten im Typo3-Seitenbaum. Eine mögliche Operation ist eben auch die Indizierung, wobei crawler die Möglichkeiten von indexed_search signifikant erweitert.

- *Indizierung durch Cron-Job*

Der Indizierungsprozess kann mit Hilfe eines externen Cron-Jobs in Linux bzw. eines Tasks in Windows aufgerufen werden und ist dadurch unabhängig vom Seitenzugriff im Frontend. Ein Vorteil ist die Aktualität. Der Index wird jede Stunde überprüft und gegebenenfalls aktualisiert.

- *Erweitertes Indizierungskonzept*

crawler erweitert das Konzept der Indizierung und dehnt es auch auf andere Bereiche als nur die Seiten und die dort verlinkten Dateien aus. Dazu bietet crawler ein zusätzliches Content Element, die „Indexing Configuration“.

Eine Indexing Configuration enthält Informationen, was wann wie oft indiziert werden soll und speichert die Ergebnisse der Indizierung. crawler bietet folgende Typen der Indexing Configuration:

- *Indexing of page tree*

Dieser Typ entspricht der normalen Indizierung des Seitenbaumes, allerdings kann man die Startseite der Indizierung angeben und zusätzlich wie tief die Indizierung im Seitenbaum durchgeführt werden soll. So können die durchsuchbaren Seiten beschränkt werden.

- *Filepath on server*

crawler erlaubt die Angabe von Ordnern auf dem Server, deren Inhalte indiziert werden sollen. So können auch Dokumente, Präsentationen, etc. indiziert werden, welche nicht direkt auf einer Webseite verlinkt werden.

- *Database records*

Eine indexing configuration dieses Typs erlaubt die Indizierung jeglichen Datenbankinhaltes, wobei auch die zu indizierenden Felder konfiguriert werden können. Dieser Typ wird z.B. zur direkten Indizierung der Newseinträge in tt_news benutzt.

- *External URL*

Auch externe Webseiten können mit Hilfe von crawler indiziert werden.

- *Meta configuration*

Dieser Typ durchsucht andere Indexing Configurations.

Damit crawler funktioniert, muss zusätzlich zu den Indexing Configurations in der PAGE-TSconfig der Root-Seite mindestens ein „paramSet“ definiert werden.

Dieses paramSet enthält neben der zu erledigenden Aufgabe auch die Domain bzw. URL der zu indizierenden Seite.

Ein paramSet muss zumindest folgende Bestandteile haben:

```
tx_crawler.crawlerCfg.paramSets{
    std.procInstrFilter = tx_indexedsearch_reindex
    std.baseUrl = http://security.fim.uni-linz.ac.at/
}
```

Mit der oben angeführten Konfiguration indiziert crawler alle Seiten, die in der Form *<Domain>/index.php?id=x* aufzurufen sind, wobei *x* für die jeweilige Seiten-ID steht. crawler iteriert automatisch durch alle gewünschten Seiten-IDs.

Seiten, die zusätzliche Parameter erfordern, wie etwa Seiten in anderen Sprachen, können ebenfalls indiziert werden. Dies erfordert zusätzliche Konfiguration für crawler. Dieser benötigt die Information, aus welcher Datenbanktabelle die zusätzlichen Parameter stammen und wie sie zusammengesetzt werden, um eine Seite in der Zweitsprache aufzurufen. Im folgenden Beispiel werden alle URLs zusätzlich mit dem Sprachparameter *L* aufgerufen, wobei crawler die Werte von *L* aus dem Feld „sys_language_uid“ aus der Tabelle „pages_language_overlay“ bezieht.

```
tx_crawler.crawlerCfg.paramSets {
    std = &L=[|_TABLE:pages_language_overlay;_FIELD:sys_language_uid]
}
```

Diese Möglichkeit kann man sich zu Nutze machen, um so alle möglichen Seiten, die von News-, Foren- oder selbstentwickelten Extensions erzeugt werden, zu indizieren und damit durchsuchbar zu machen.

Schließlich muss crawler noch als Cron-Job (Linux) bzw. als Task (Windows) eingetragen werden und regelmäßig gestartet werden. Dazu enthält die Extension eine PHP-Datei, welche bei Aufruf alle konfigurierten Indizierungsvorgänge anstößt. Damit das Skript auch von außen funktioniert, ist die Erstellung eines Backendbenutzers, „cli_crawler“, mit beliebigem Passwort und ohne Rechte notwendig.

6.6.5. **kj_imagelightbox2**

Diese Extension sorgt für eine professionellere Präsentationsmöglichkeit von Bildern. Standardmäßig wird in Typo3 ein angeklicktes Vorschaubild einfach in einem neuen Browserfenster geöffnet. Die Imagelightbox sorgt dafür, dass das Bild in einem mit transparentem Schwarz hinterlegten Fenster mit einem JavaScript-Effekt geöffnet wird. Weiters werden die Funktionen „Speichern“, sowie „Drucken“ angeboten. Möglich ist auch die Verknüpfung mehrerer Bilder zu einer Diashow.

6.6.6. **mailformplus, kt_student_file_system**

Ein zentraler Bestandteil des Security-Portals ist die Möglichkeit, Studentarbeiten auf der Webseite mit Hilfe eines „Mini-Workflow“ zu veröffentlichen.

Dieser Vorgang besteht aus mehreren Teilschritten, welche von zwei Extensions durchgeführt werden: der bereits vorhandenen Extension „mailformplus“ und der selbst erstellten Extension „kt_student_file_system“.

- *Upload der Datei*

Jede Lehrveranstaltungsseite bietet ein Uploadformular an. Mit dem Formular können die Dateien hochgeladen werden. Dieser Schritt wird vollständig durch die bereits vorhandene Extension mailformplus unterstützt.

- *Freischaltung/Löschung der Datei*

Um Missbrauch wirkungsvoll unterbinden zu können, muss jede Datei vor der Freischaltung durch einen Verantwortlichen geprüft werden. Dieses Freischaltungssystem ist Email-basiert, d.h. der Verantwortliche bekommt eine Email und kann über Links in der Email die Datei sperren oder zum Download freigeben. Die Email wird mit Hilfe der Extension mailformplus versendet. Der eigentliche Freischalte- bzw. Löschvorgang wird von der Extension kt_student_file_system erledigt.

- *Präsentation der Dateien auf der Webseite*

Ist eine Datei freigeschaltet, so soll sie natürlich auch heruntergeladen werden können. Die Präsentation der Dateien geschieht auf verschiedenen Seiten auf verschiedene Art und Weise. Auf dem Hauptportal etwa werden die neuesten Dateien des gesamten Portals angeboten, während auf der Lehrveranstaltungsseite alle Dateien der Lehrveranstaltung in Listenform angezeigt werden.

Diese Funktionalität wird ebenfalls von der Extension kt_student_file_system erfüllt.

6.6.6.1. Mailformplus

Features

Die Erweiterung mailformplus bietet folgende Features, welche für das Upload-System benötigt werden:

- Template-basierte Erstellung von Formularen
- Dateupload mit Beschränkungsmöglichkeiten auf Dateitypen bzw. -größe,
- Automatische Umbenennung jeder hochgeladenen Datei zur Vermeidung von doppelten Dateinamen in der Form <Zeit-Hashwert>_<UID des DB-Eintrags>_<PID der LVA-Seite>_<Dateiname>
- Konfigurierbare Validierung aller Eingabefelder
- Automatisches Speichern der Eingabewerte in eine Datenbanktabelle
- Möglichkeit zum Aufruf einer benutzerdefinierten Funktion nach dem Senden des Formulars

Konfiguration

Die Konfiguration von mailformplus passiert an drei verschiedenen Stellen:

- *Plugin*

Das Frontend-Plugin wird über den Seiteneditor an der gewünschten Stelle der Seite eingefügt. Im Plugin selbst muss nur das zu verwendende Template-File angegeben werden.

Zusätzlich können noch einige E-Mail Einstellungen konfiguriert werden, wobei für die Zwecke des Security-Portals nur die Einstellung der Empfängeradresse des Bestätigungsemails relevant ist. An diese Adresse wird später das Email mit der Bitte um Freigabe bzw. Löschung geschickt.

- *Template*

Das Template ist eine HTML-Vorlage jenes Formulars, das in der Webseite später angezeigt wird. Das Template enthält Marker der Form ###Marker###, welche durch mailformplus bzw. das zugehörige Template ersetzt werden. Im Template sind auch einige versteckte Felder integriert, welche zur Erstellung der Email an den für die Freischaltung Verantwortlichen benötigt werden (z.B. wird die Seiten-ID im Marker ###PID### gespeichert).

Unten ist das Haupttemplate gekürzt wiedergegeben.

Einer der großen Vorteile von mailformplus ist eben die Validierung, die vor dem Absenden des Formulars durchgeführt wird. Alle Fehler werden gesammelt im Marker

###ERROR### dargestellt, jeder Fehler separat noch einmal beim betroffenen Feld (###error_feldname###). Damit im Falle eines Fehlers die Felder nicht immer neu eingegeben werden müssen, werden die eingegebenen Werte gespeichert und wieder eingesetzt (###value_feldname###).

```
<form          name="Formular"          method="post"          action="###URL###"
enctype="multipart/form-data">

  <input type="hidden" name="id" value="###PID###">
  <input type="hidden" name="seitentitel" value="###SEITENTITEL###">

  ###ERROR###

      <p class="upload_form">Titel:<br/>
      ###error_titel###
      <input type="text" name="titel" value="###value_titel###">
</p>
<p class="upload_form">Datei: <br />
  ###error_file#####value_file###
<input      type="file"      name="file"      size="20"      maxlength="1000000"
accept="*.pdf, *.doc, *.docx, *.ppt, *.pptx"></td>
</p>
</form>
```

- *TypoScript – Konfiguration*

Einige Funktionen in mailformplus können über TypoScript konfiguriert werden. So lassen sich händisch in ein Template eingefügte Marker, wie oben der Seitentitel, durch TypoScript ersetzen.

```
temp.seitentitel.value = {page:title}
plugin.tx_thmailformplus_pi1.markers.SEITENTITEL < temp.seitentitel
```

Die automatische Speicherung der Eingabewerte in der Datenbank muss ebenfalls in TypoScript aktiviert werden. Damit diese Speicherung funktioniert, müssen die Felder den gleichen Namen wie die korrespondierende Tabellenspalte in der Datenbank besitzen. Ist dies nicht der Fall, so müssen die Feldnamen zu den Tabellennamen gemappt werden.

```
plugin.tx_thmailformplus_pi1.saveDB = 1
plugin.tx_thmailformplus_pi1.saveDB{
  dbTable = tx_ktstudentfilesystem_kt_uploaded_files
  fields.id.mapping = pid
}
```

Schließlich müssen noch die Validierungseinstellungen getroffen werden. mailformplus bietet hier weitreichende Einstellungsmöglichkeiten. Exemplarisch sind die Einstellungen für das Feld „email“ und auch für das Feld „file“ angegeben.

```
email {
    errorCheck = required
    errorCheck = email
    errorText=<li>Bitte korrekte e-Mail Adresse angeben!</li><br />
}
file {
    errorCheck = required
    numberOfAllowedFiles = 1
    fileTypes = pdf,doc,docx,ppt,pptx
    fileSize = 10000000
    fileUpload = uploads/tempfiles/
    errorText = <li>Bitte upzuladende Datei kontrollieren!</li><br />
}
```

Auch die Einstellungen für die englische Sprache müssen getroffen werden, wobei hier natürlich nur der Wert von errorText geändert werden muss. Exemplarisch für das Feld email:

```
[globalVar = GP:L = 1]
plugin.tx_thmailformplus_pi1.fieldConf.email.errorText=
    <li>Please fill in a valid email address!</li><br />
[global]
```

mailformplus ermöglicht die Ausführung einer benutzerdefinierten Funktion nach dem Abschicken des Formulars. Dies wird im konkreten Fall dazu verwendet, um die Email an den Verantwortlichen für die Freischaltung zu erstellen und zu senden.

```
plugin.tx_thmailformplus_pi1.saveUserFunc =
    fileadmin/scripts/MailScript.php:user_sendMail->main
```

In der Standardversion wird diese Funktion noch vor den Datenbankoperationen von mailformplus aufgerufen. Die gewünschte Funktionalität der Benutzerfunktion erfordert aber im konkreten Fall, dass bereits Zugriff auf die Werte der Datenbank besteht. Die Änderung des Quellcodes der Extension ist somit unerlässlich.

Der Quellcode der Datei *pi1/class.tx_thmailformplus_pi1.php* wird wie folgt geändert: Die Zeilen 1625 bis 1631, welche den Aufruf der benutzerdefinierten Funktion enthalten, werden auskommentiert und stattdessen ab Zeile 2156 nach den Datenbankoperationen eingefügt.

Benutzerdefinierte Funktion

Zusätzlich muss auch die benutzerdefinierte Funktion erstellt werden. Diese soll dazu verwendet werden, den für das Review einer hochgeladenen Datei Zuständigen über den Upload zu informieren.

Mailformplus speichert alle neuen Dateien in einem nicht direkt aus dem Internet zugänglichen Ordner. Konkret kann auf diesen Ordner nur vom Server selbst, also von der IP-Adresse 127.0.0.1 zugegriffen werden. Dies erlaubt den Zugriff durch Typo3-Extensions, weil diese ja lokal auf dem Server laufen.

Um die Begutachtung der Datei zu ermöglichen, ist in der Email ein Link vorhanden, der auf das Download-Plugin der Erweiterung `kt_student_file_system` verweist. Dieses Plugin erwartet den Hash-Wert des Dateinamens einer hochgeladenen Datei und bietet diese, so vorhanden, zum Download an.

Da der erwähnte Hash-Wert aus dem von mailformplus generierten Dateinamen berechnet wird, ist es unmöglich, den Hash-Wert ohne Kenntnis der Email zu erlangen. Der Missbrauch des Systems ist damit praktisch unmöglich.

Weiters sollen direkt in der Email Links die Freischaltung bzw. Löschung einer Datei ermöglichen. Das Ziel der Links ist dabei jene Seite, die das Freischalte-Plugin der Erweiterung `kt_student_file_system` enthält. Dieses Plugin erwartet als GET-Parameter eine File-ID. Diese File-ID enthält den Hash-Wert des Dateinamens gefolgt von der Information, ob die Datei freigeschaltet oder gelöscht werden soll. Zusätzlich enthält sie noch die Seiten-ID jener Seite, der die Datei zugeordnet ist. Folgender Link ist exemplarisch für die Freischaltung einer Datei:

<http://140.78.100.10/index.php?id=41&fid=82777c6cf1ce58ecfaceb7401b064b00056>

Die Löschung dieser Datei würde über folgenden Link erfolgen:

<http://140.78.100.10/index.php?id=41&fid=82777c6cf1ce58ecfaceb7401b064b00156>

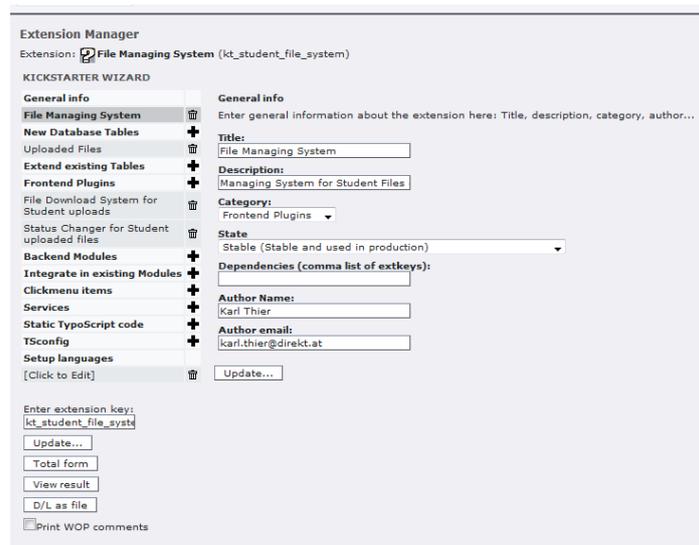
6.6.6.2. `kt_student_file_system`

An Hand der Extension `kt_student_file_system` soll in einem konkreten Beispiel die Erstellung einer Extension gezeigt werden. Das Grundgerüst wird mit Hilfe von kickstarter erstellt, die Programmlogik inklusive Sprachunterstützung wird anschließend händisch hinzugefügt.

Wiewohl die vorgestellte Extension nicht die gesamte Bandbreite an Möglichkeiten von kickstarter ausnutzt, so wird doch ein guter Überblick gegeben.

kickstarter

Begonnen wird im kickstarter mit der Eingabe eines „Extension-Keys“, sowie einiger Metainformationen wie Autor, Email-Adresse, Kategorie und Status der Extension.



The screenshot shows the 'Extension Manager' interface for the 'File Managing System' extension. The interface is divided into two main sections: a left sidebar with a list of actions and a main content area for editing the extension's details. The sidebar includes options like 'General info', 'New Database Tables', 'Uploaded Files', 'Extend existing Tables', 'Frontend Plugins', 'Backend Modules', 'Integrate in existing Modules', 'Clickmenu items', 'Services', 'Static TypoScript code', 'TSconfig', and 'Setup languages'. The main content area is titled 'KICKSTARTER WIZARD' and contains a 'General info' section with fields for 'Title', 'Description', 'Category', 'State', 'Dependencies', 'Author Name', and 'Author email'. The 'Title' field contains 'File Managing System', 'Description' contains 'Managing System for Student Files', 'Category' is set to 'Frontend Plugins', and 'State' is set to 'Stable (Stable and used in production)'. The 'Author Name' is 'Karl Thier' and the 'Author email' is 'karl.thier@direkt.at'. There is an 'Update...' button at the bottom of the main content area. At the bottom of the interface, there is a section for 'Enter extension key:' with the value 'kt_student_file_system' and buttons for 'Update...', 'Total form', 'View result', and 'D/L as file'. A checkbox for 'Print WOP comments' is also present.

Abbildung 23: Oberfläche von Kickstarter

Die Informationen zu den hochgeladenen Dateien werden in der Tabelle `tx_ktstudentfilesystem_kt_uploaded_files` gespeichert. Beachtet werden muss, dass 16 Felder in jeder Typo3-Tabelle angelegt werden. Dazu zählen „UID“ als eindeutiger Index, „PID“ als Seiten-ID bzw. auch „fe_group“, in der Benutzerberechtigungen für das Frontend gespeichert werden. Diese 16 Feldnamen sind reserviert und dürfen nicht noch einmal verwendet werden. Die Feldtypen der benutzerdefinierten Felder entsprechen den Feldtypen im Typo3-Backend. Typo3 mappt diese automatisch auf die richtigen Datenbank-Feldtypen.

Folgende benutzerdefinierte Felder werden zur Tabelle hinzugefügt:

- *mail_teacher, title, email, filename, ip, semester, hash*

Diese Felder sind vom Typ „String input“. Speichert die Mail-Adresse des für die Freischaltung Verantwortlichen, sowie email und IP des Absenders und den Titel der hochgeladenen Datei. Weiters wird der Dateiname und das Semester, dem die Datei zugeordnet ist, gespeichert. Besonders wichtig ist der Hash-Wert, mit diesem wird nämlich die Freischaltung bzw. Löschung der Datei vorgenommen.

- *kurzfassung, abstract, tags*
Sind vom Typ „Text area“ und enthalten deutsche und englische Kurzfassungen sowie die mit der Datei assoziierten Tags.
- *datetime*
Ist vom Typ “date and time” und speichert den Upload-Timestamp.
- *hide, portalid*
„hide“ enthält die Information, ob die Datei freigeschaltet ist. Im Feld „portalID“ wird die Seiten-ID des mit einer Lehrveranstaltung assoziierten Themenportals gespeichert.

Die vorgestellte Extension `kt_student_file_system` bietet drei Frontend-Plugins an. Zum Ersten ist dies jenes Plugin, mit dem die Dateien auf den Lehrveranstaltungsseiten sowie auf den Portalseiten präsentiert und zum Download angeboten werden. Zum Zweiten wird auch das Plugin zur Freischaltung bzw. Löschung von hochgeladenen Dateien durch die Verantwortlichen im Rahmen dieser Extension erstellt. Das dritte Plugin fungiert als Download-Skript, denn aus Sicherheits- und Usabilitygründen werden die freigegebenen Dateien nicht direkt verlinkt.

Bei der Erstellung eines neuen Plugins muss man neben dem Namen und einer optionalen Beschreibung zusätzlich angeben, wo im Backend das Plugin später wählbar sein soll. Dabei ist für klassische Plugins, wie die zu erstellenden sein sollen, die default-Einstellung „Add to ‚Insert Plugin‘ list in Content Element“ sinnvoll.

Welche Funktionen diese Plugins erfüllen sollen und welche Inhalte sie ausgeben, ist für den kickstarter vollkommen unerheblich, denn die Programmlogik wird erst später eingefügt. Die drei Plugins erhalten die Namen „File Download System for Student Uploads“, „Status Changer for Student uploaded files“ und „File Download Plugin“.

Nachdem sämtliche Inhalte des Security-Portals zweisprachig sein sollen, wird im Feld `Setup languages` zusätzlich zur bereits eingestellten Sprache Englisch noch Deutsch ausgewählt.

Anschließend wird durch Klick auf „View result“ und „Write“ der generierte Programmcode direkt ins lokale Extension Repository eingefügt, sowie im nächsten Schritt installiert. Ab diesem Zeitpunkt ist auch die Tabelle in der Datenbank vorhanden, sodass bereits Dateien hochgeladen werden können.

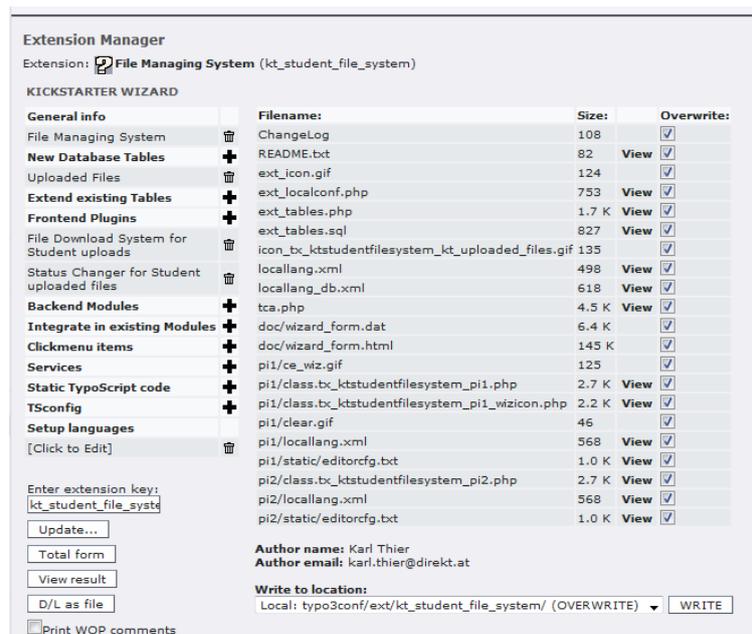


Abbildung 24: Die von Kickstarter erzeugten Dateien

Damit die Plugins tatsächlich eingesetzt werden können, muss nun noch die Programmlogik ergänzt werden.

Die von Typo3 zum Rendern der Frontendausgabe der Plugins benutzten PHP-Dateien `class.tx_ktstudentfilesystem_pi1.php` bzw. `[...]pi2.php` liegen im Ordner `/typo3conf/ext/xml_ttnews_import_extended/cm1/` bzw. `[...]cm2/`. Sie enthalten bereits die von Typo3 zum Rendering aufgerufene Main-Methode und erzeugen schon Dummy-Inhalte.

Programmlogik von File Download System for Student uploads

Das Plugin zur Präsentation der Dateien soll an drei verschiedenartigen Stellen der Seite eingebunden werden. Zum Ersten natürlich auf den Lehrveranstaltungsseiten, wo alle Dateien, die mit dem Formular dieser Lehrveranstaltung hochgeladen wurden, angezeigt werden sollen. Zum Zweiten soll auf den Portalseiten sowie zum Dritten auch auf der Hauptseite die Möglichkeit bestehen, die zuletzt hochgeladenen Dateien herunterzuladen. Auf den Portalseiten werden nur Dateien von Lehrveranstaltungen angeboten, die mit dem gewählten Portal assoziiert sind. Auf der Hauptseite wird die Auswahl von allen vorhandenen Dateien getroffen.

Die Herausforderung ist, zu eruieren, welche Tiefe im Seitenbaum die gerade aufgerufene Seite besitzt.

Ist diese Tiefe eins, so befindet man sich auf der Hauptseite und das Plugin muss die zuletzt hochgeladenen Dateien der gesamten Webseite ausgeben. Ist sie zwei, so befindet man sich auf einer Portalseite. Dann werden die zuletzt hochgeladenen Dateien dieses Portals

präsentiert. Ist sie jedoch drei oder höher, wird gerade eine Lehrveranstaltungsseite angezeigt und das Plugin muss alle Dateien der Lehrveranstaltung anzeigen.

Die Funktion `$GLOBALS["TSFE"]->sys_page->getRootLine($GLOBALS["TSFE"]->id)` liefert ein Array mit dem Seitenpfad zur aktuellen Seite. Die Anzahl der Elemente im Seitenpfad entspricht demnach der Tiefe einer Seite im Seitenbaum. Mit dieser Information lässt sich feststellen, welche Art der Ausgabe vonnöten ist.

Ob eine Datei angezeigt wird, hängt vom Wert des Feldes `hide` in der Tabelle `ktstudentfilesystem_kt_uploaded_files` ab. Bei jeder neu hochgeladenen Datei, die noch durch den Zuständigen überprüft und dann freigegeben werden muss, ist dieser Wert 1. Damit wird verhindert, dass diese Datei angezeigt und damit heruntergeladen werden kann.

Nach der Freigabe der Datei wird `hide` auf 0 gesetzt und die Datei somit auch auf der Webseite angezeigt.

Programmlogik von File Status Changer for Student uploaded files

Aufgabe dieses Plugins ist die Freischaltung bzw. Löschung einer Datei. Wird die Webseite aufgerufen, die dieses Plugin enthält, so muss in der GET-Variable die Variable „fid“ definiert sein. Diese Variable besteht aus dem Hash-Wert des Dateinamens, gefolgt von 1 für Löschen oder 0 für Freischalten, gefolgt von der Seiten-ID jener LVA-Seite, der die Datei zugeordnet ist.

- *Löschung*

Soll die Datei gelöscht werden, so wird der Eintrag der Datei in der Tabelle `tx_ktstudentfilesystem_kt_uploaded_files` gelöscht. Auch die Datei selbst wird in diesem Fall vom Server gelöscht.

- *Freischaltung*

In diesem Falle wird die Variable „hide“ in der Tabelle auf 0 gesetzt, sodass das Download-Plugin die Datei auch anzeigt.

Jede hochgeladene und freigeschaltete Datei soll indiziert und somit vollständig durchsuchbar gemacht werden. crawler ermöglicht die Indizierung von Dateien nur Ordnerweise, sodass es notwendig ist, freigeschaltete Dateien in einen anderen Ordner zu verschieben. Die Dateien werden vom Ordner `uploads/tempfiles`, der nur für die IP-Adresse `127.0.0.1` zugänglich ist, in den Ordner `fileadmin/studentfiles` verschoben. Dieser Ordner ist öffentlich zugänglich und wird automatisch von `indexed_search` bzw. crawler indiziert.

Programmlogik von File Download Plugin

Das „File Download Plugin“ fungiert als Download-Skript für die auf *security.fim.unilinz.ac.at* hochgeladenen Dateien. Die Abwicklung der Downloads über ein Skript ist auf Grund folgender Tatsachen empfehlenswert bzw. notwendig:

- *Review der Dateien*

Die Dateien sind vor der Freischaltung nicht für die Öffentlichkeit zugänglich. Trotzdem muss eine Möglichkeit für den Reviewer bestehen, die Datei zu begutachten. Über das Skript ist jede Datei sofort nach dem Upload verfügbar, vorausgesetzt, man kennt den richtigen Hash-Wert des Dateinamens. Da der Dateiname aber von mailformplus geändert und unter anderem mit einem Zeitstempel versehen wird, ist es ausgeschlossen, den Hash-Wert herauszufinden.

Der Wert ist jedoch Teil der Email an den Reviewer.

- *Änderung des Dateinamens beim Download*

Auch nach der Freischaltung der Datei ist die Verwendung eines Plugins sinnvoll. Wie oben erwähnt, ändert mailformplus die Dateinamen der hochgeladenen Dateien nach einem bestimmten Schema. Das Plugin ermöglicht es, den ursprünglichen Dateinamen einer Datei für den Download wiederherzustellen, sodass die Änderungen von mailformplus für den Benutzer nicht sichtbar sind.

- *Erzwungenes Herunterladen einer Datei*

Durch die Verwendung der PHP-Methode „readfile“, sowie der Konfiguration als „Attachement“ im Header der ausgegebenen Datei wird der Browser gezwungen, das Downloadfenster anzuzeigen, während bei manchen Dateitypen bei Verwendung einer Direktverlinkung browserinterne Fehler auftreten können.

Bei der Freischaltung werden die Dateien durch den „File Status Changer“ in einen anderen Ordner verschoben, damit die Durchsuchbarkeit gegeben ist. Das Download-Skript erkennt daher auch, ob eine Datei bereits freigegeben ist und greift automatisch auf den richtigen Speicherort zu.

6.6.7. kt_kusss_date_export

Um redundante Datenhaltung und einen hohen Administrationsaufwand zu vermeiden, sollen die Lehrveranstaltungstermine aus dem KUSSS exportiert und direkt in die Webseite eingebunden werden. Die Extension, die diese Aufgabe erfüllt, ist eine vollständige Eigenentwicklung. Zur Erstellung der Datei- und Ordnerstruktur wird die Extension kickstarter benutzt.

6.6.7.1. Voraussetzungen

Das KUSSS-System bietet Funktionen, um einzelne Daten zu Lehrveranstaltungen, zu Lehrenden, etc. abzurufen. Die folgende Funktion ermöglicht die Abfrage aller Daten zu einer Lehrveranstaltung mit einer gegebenen Lehrveranstaltungsnummer:

```
select-course.action?courseid=<LVA-NR>&schedule=true&type=xml
```

Die Funktion liefert die Termine in folgendem XML-Format:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<courses><course id="351098">...<schedule>
  <course-date>
    <date>04.03.2008</date>
    <time-begin>13:45</time-begin>
    <time-end>15:15</time-end>
    <location>HS 12 </location>
    <theme></theme>
  </course-date>
  <course-date>
    ...
  </course-date>
  ...
</schedule></course></courses>
```

6.6.7.2. Die Implementierung

Als Parameter der KUSSS-Funktion wird die Lehrveranstaltungsnummer der aktuellen Lehrveranstaltungsseite benötigt.

Der Inhalt dieser Seiten ist in ein mit Hilfe von TemplaVoila erstelltes Flexible Content Element eingebettet. Wie alle Inhalte in Typo3 befinden sich auch die Flexible Content Elements mitsamt ihren Inhalten in XML-Form in der Tabelle tt_content der Datenbank.

Mit Hilfe der Abfrage

```
$query = $GLOBALS['TYPO3_DB']->SELECTquery
('tx_templavoila_flex', 'tt_content', 'pid='.$GLOBALS['TSFE']->id.' and
tx_templavoila_ds="5");
```

werden Struktur und Inhalt des Lehrveranstaltungstemplates (tx_templavoila_ds="5") der aktuellen Seite (\$GLOBALS["TSFE"]->id) aus der Datenbank geladen. Der Inhalt und die Struktur des Templates liegen als XML-String vor. Ab Version 5 enthält PHP die Erweiterung SimpleXML. Diese Bibliothek ermöglicht einfache Operationen auf XML-Strings bzw. XML-Dateien.

Im konkreten Fall muss die Lehrveranstaltungsnummer aus dem der Datenbank entnommenen Flexible Content Element extrahiert werden. Folgende zwei Codezeilen sind dazu nötig:

```
$xmlvo = simplexml_load_string($text['tx_templavoila_flex']);  
$datum = $xmlvo->data->sheet->language->field[2]->value;
```

Die Variable `$text` enthält das Resultat der SQL-Abfrage, also das Flexible Content Element der aktuellen Lehrveranstaltungsseite.

Nachdem der in der Lehrveranstaltungsnummer enthaltene Punkt gelöscht wurde, kann die Abfrage an das KUSSS-System gestartet werden. Das Resultat wird wiederum mittels SimpleXML geparkt und weiterverarbeitet.

Um die Übersichtlichkeit der Ausgabe zu gewährleisten, wird nur der nächstfolgende Termin auf der Seite direkt angezeigt. Der JavaScript-Slider ermöglicht eine Ansicht aller Termine eines Semesters.

6.6.8. **kt_security_widgets**

Um die Attraktivität der Seite weiter zu erhöhen, werden in der Leiste links nützliche Zusatzinformationen in Form von Widgets eingebunden.

6.6.8.1. Implementierung

Zur Realisierung der Widgets wird die JavaScript-Bibliothek „Docking Boxes“ von Brothercake[BRO-08] verwendet.

Die Docking Boxes sind in einer dbx-group eingebettet. Eine Box besteht aus dem dbx-handle sowie dem dbx-content. Im dbx-handle befindet sich neben dem Titel auch ein Bild mit einem dahinterliegenden Link. Durch Klick auf diesen Link kann mit Hilfe von JavaScript die Sichtbarkeit des Inhaltsbereiches gesteuert werden.

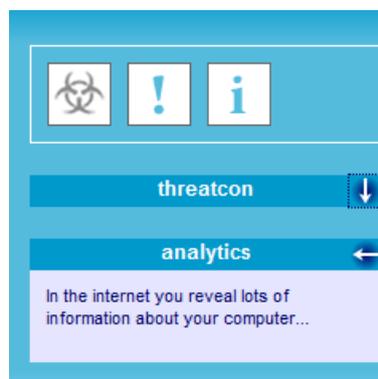


Abbildung 25: Security-Widgets
Linkes Icon: Box versteckt
Mitte: Box eingeklappt
Rechts: Box ausgeklappt

Der Status des Inhaltsbereiches einer Box wird dabei mit Hilfe eines Cookies gespeichert, sodass beim nächsten Aufruf der Seite die Box geöffnet bzw. geschlossen ist.

Die Reihenfolge der Boxen einer dbx-group kann per Drag&Drop geändert werden.

Zusätzlich müssen für die `kt_security_widgets` folgende Features implementiert werden:

- *Ein- und Ausblenden der gesamten Box*

Die Funktion zum Ein- und Ausblenden des Inhaltsbereiches ist im JavaScript bereits vorhanden. Zur Änderung des Zustandes verändert diese die Klassenattribute des betroffenen Inhaltsbereiches, wodurch sich die CSS-Formatierung ändert. Der Zugriff auf das richtige HTML-Element erfolgt relativ zum DOM-Baum, da sich der Link zum Ein- bzw. Ausblenden relativ zum Inhaltsbereich immer an der gleichen Stelle befindet.

Prinzipiell funktioniert die Steuerung der Anzeige der Gesamtbox nach dem gleichen Prinzip. Um eine Box auszublenden, werden die Klasseneigenschaften und damit die CSS-Formatierung geändert. Dazu wird eine zusätzliche Ebene in Form eines zusätzlichen DIV-Tags eingeführt. Dieser Tag wird über seine ID angesprochen, da die Links zur Steuerung des Anzeigeverhaltens der Gesamtbox in eine externe Box des Typs `dbx-control` ausgelagert sind.

Zusätzlich wird auch der Zustand der Gesamtbox in den Cookies gespeichert.

- *Sliding-Effekt statt einfachem Ein- und Ausblenden*

Aus Gründen der durchgehenden Gestaltung der Webseite wird auch das Ein- und Ausblenden der Security-Widgets mit der oben erwähnten Sliding-Bibliothek erweitert, sodass die Boxen bei einem Zustandswechsel auf- bzw. zuklappen und sich zusätzlich die Transparenz ändert.

- *Inhalte sollen per Ajax (nach)geladen werden*

Die security-Widgets sind als eine Art „Gimmick“, als Ergänzung zu den Hauptinhalten von `security.fim.uni-linz.ac.at` gedacht, werden aber auf jeder Seite angezeigt und aus Gründen der Aktualität auch mit jedem Seitenaufruf neu geladen. Um die Ladezeiten der eigentlichen Seite nicht zu beeinträchtigen, werden daher die Inhalte der security-Widgets erst nach dem vollständigen Laden der Seite per „Ajax“ geladen. Die Ajax-Anfragen werden nach dem vollständigen Laden einer Seite durch das `onload`-Ereignis von JavaScript ausgelöst. Wie in Abschnitt 4.7.8 bereits eingehend erläutert, werden die Ajax-Abfragen mit Hilfe der Funktionen der JavaScript-Bibliothek „Prototype“ durchgeführt. Diese Funktionen greifen über den

Typo3-eigenen eID-Mechanismus auf PHP-Skripts der Erweiterung zu, welche dann die eigentlichen Inhalte generieren.

6.6.8.2. Die Widgets

Beim Start der Webseite sind drei Widgets im Container enthalten:

- *Bedrohungsstatus*

Der Bedrohungsstatus zeigt die aktuelle Sicherheitslage im Internet an Hand eines vierstufigen Ampelsystems. Die Information über die Bedrohungslage stammt vom SANS Internet Storm Center, einem Institut, das sich mit Netzwerk- und Internetsicherheit sowie Audits, Zertifizierung und Forschung in diesem Bereich beschäftigt.

- *Aktuelle Viren*

Dieses Widget liefert Informationen zu den aktuellsten Viren. Die Informationen dazu stammen aus einem RSS-Feed von www.viruslist.com. Dieser Feed wird mittels SimpleXML geparst, die aktuellsten fünf Einträge werden im Widget angezeigt.

- *Gläserner Nutzer*

Das dritte Widget soll dem Benutzer demonstrieren, welche Informationen er alleine durch die Präsenz auf einer Webseite preisgibt. Die verschiedenen Informationen werden dabei mittels PHP- und JavaScript-Funktionen eruiert und mittels eines JavaScript-Faders präsentiert. Genaue Informationen über die Inhalte und Funktionsweise des Widgets sind im folgenden Kapitel zu finden.

7. Integration des 10h Praktikums „Gläserner Benutzer“ in *security.fim.uni-linz.ac.at*

Ein Teil der Diplomarbeit besteht darin, die Ergebnisse des 10-stündigen Praktikums in *security.fim.uni-linz.ac.at* einzubinden.

7.1. Aufgabenstellung

Im 10h Praktikum wurden Möglichkeiten untersucht und bearbeitet, die ein Betreiber einer Webseite bzw. eines Servers hat, um auf legalem Wege möglichst viele Informationen über einen Besucher seiner Webseite zu erlangen.

Dabei sollten gängige Webtechnologien, wie HTML, CSS, JavaScript, Java, etc. untersucht werden. Die erarbeiteten Ergebnisse sollten in einer Demonstrationswebseite veröffentlicht

werden. Dieser Online-Sicherheitstest wird nun auf *security.fim.uni-linz.ac.at* veröffentlicht und soll so die Qualität und den Funktionsumfang des Portals weiter steigern.

7.2. Möglichkeiten der Informationssammlung

In diesem Kapitel werden die gefundenen Möglichkeiten zur Informationssammlung kurz umrissen. Gegliedert ist das Kapitel dabei nach den verwendeten Webtechnologien.

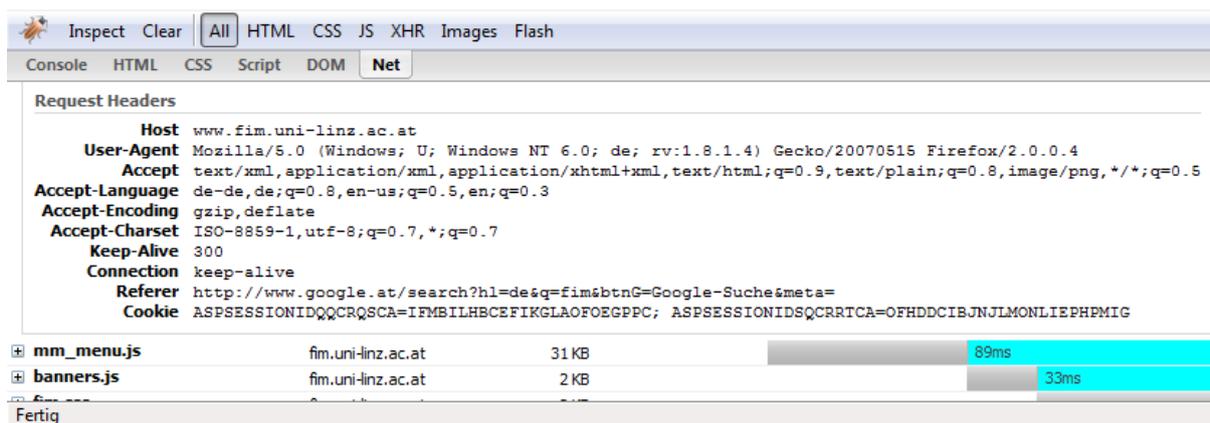
7.2.1. TCP/IP-Verbindung

Bereits die einfache Verbindungsaufnahme per TCP/IP gibt einige Daten des Benutzers preis. Mit Hilfe der IP-Adresse kann der verwendete Computer identifiziert werden. Über IP-Adressdatenbanken ist die Zuordnung dieser IP-Adresse zu einem Land bzw. zu einer Region möglich. Verwendet wird dies beispielsweise bei länderspezifischer Werbung.

Ist die IP-Adresse in einer „WHOIS-Datenbank“ gespeichert, kann eventuell sogar der Besucher selbst identifiziert werden, meist ist aber nur der Provider, der die jeweilige IP-Adresse vergibt, zu finden.

Sämtliche dieser Ansätze hängen von der IP-Adresse ab. Diese kann durch dynamisch vergebene Adressen verfälscht werden, Internetbenutzer können anonymisiert über Proxies bzw. Anonymisierungsnetzwerke im Internet surfen und so diese Zuordnung erschweren oder sogar unmöglich machen.

7.2.2. HTTP-Request



The screenshot displays the 'Request Headers' section of a web browser's developer console. The headers are as follows:

- Host:** www.fim.uni-linz.ac.at
- User-Agent:** Mozilla/5.0 (Windows; U; Windows NT 6.0; de; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4
- Accept:** text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
- Accept-Language:** de-de,de;q=0.8,en-us;q=0.5,en;q=0.3
- Accept-Encoding:** gzip,deflate
- Accept-Charset:** ISO-8859-1,utf-8;q=0.7,*;q=0.7
- Keep-Alive:** 300
- Connection:** keep-alive
- Referer:** http://www.google.at/search?hl=de&q=fim&btnG=Google-Suche&meta=
- Cookie:** ASPSESSIONIDQQCRQSCA=IFMBILHBCFIRGLAOFQEGPPC; ASPSESSIONIDSQCRRTCA=OFHDDCIBJNJLMLONLIEPHPMIG

Below the headers, a table shows the loading of resources:

Resource	Size	Time
mm_menu.js	31 KB	89ms
banners.js	2 KB	33ms

Abbildung 26: HTTP-Request eines Webbrowsers

Wenn ein Browser eine Verbindung zu einem Webserver aufbaut um eine Seite abzurufen, so werden automatisch und ohne Wissen des Benutzers bestimmte Informationen übertragen:

- Name, Version und in einigen Fällen sogar Patchlevel des aufrufenden Browsers
- Das verwendete Betriebssystem bzw. die Version davon
- Die vom Browser unterstützten bzw. voreingestellten Sprachen
- Die akzeptierten Zeichensätze und Codierungen
- Der Referrer, der die zuvor besuchte Seite enthält

Einerseits sind diese Informationen erforderlich, damit der Webserver die gewünschte Webseite richtig zurückliefern kann. Des Weiteren ermöglichen die Informationen auch praktische Features, wie sprachspezifische Seiten, betriebssystemspezifische Downloadangebote und anderes mehr.

Andererseits wird durch die Übertragung von Versions- und zum Teil sogar Revisionsnummern das Ausnutzen von Schwachstellen der Produkte dieser Versionsnummern wesentlich erleichtert.

Abhilfe schaffen hier nur regelmäßige Updates von Browser und Betriebssystem. Einige Browser ermöglichen auch, die Informationen des HTTP-Requests zu konfigurieren, sodass der Webserver denkt, er habe mit einem anderen Browser auf einem anderen Betriebssystem zu tun.

7.2.3. **Java-Applets**

Prinzipiell werden Java-Applets in einer Sandbox ausgeführt, welche den Computer vor Schaden beschützen soll.

Durch die Java-Funktion `InetAddress.getLocalHost().getHostAddress()`; kann die tatsächliche IP-Adresse des Clients ausgelesen werden, wenn sich dieser z.B. in einem Firmennetzwerk hinter einem Network Address Translation-fähigen Router verbirgt. Dies alleine ist nicht bedenklich, kombiniert mit anderen Technologien kann diese Funktion jedoch unter anderem dazu benutzt werden, Netzwerke auszuspionieren.

7.2.4. **JavaScript**

JavaScript bietet sehr viele Funktionen, mit denen Informationen über den Clientcomputer abgefragt werden können. Dazu zählen Browsername und –version, Betriebssystemname und –version, Bildschirmauflösung, Farbtiefe, Cookies und installierte Plugins. Mit Hilfe von

JavaScript lassen sich sowohl Mausklicks, die Mausposition und auch Tastendrucke protokollieren.

7.2.4.1. Fingerprinting mit JavaScript

Mit Hilfe von JavaScript ist es möglich, aktive Rechner in einem Netzwerk bzw. auch Webserver zu identifizieren. Es ist sogar möglich, Router zu kompromittieren.

Die Vorgehensweise ist dabei folgende:

- *IP-Adresse vergeben?*

Das Source-Attribut eines JavaScript-eigenen Imageobjektes wird auf die zu testende IP-Adresse gesetzt. Ist an dieser IP-Adresse ein Host vorhanden, so wird ein onload bzw. ein onerror-Event ausgelöst. Ist kein Host vorhanden, so endet die Anfrage in einem Timeout.

- *Webserver vorhanden?*

Mit Hilfe eines iFrame-Objektes wird versucht, von der im 1. Schritt als aktiv identifizierten IP-Adresse HTTP-Content zu laden.

Wieder gilt: Bei Erfolg wird das onload-Event des iFrames ausgelöst, ansonsten endet die Anfrage in einem Timeout

- *Welcher Router/Server?*

Beinahe jeder Router/Server besitzt öffentlich zugängliche Bilder als graphische Elemente der Benutzeroberfläche. Diese besitzen bei Router gleicher Bauart immer die gleiche Größe und den gleichen Dateinamen.

Auf dieser Basis kann wiederum mit Hilfe eines Image-Objektes das Vorhandensein eines solchen Bildes geprüft werden und damit ein Router bzw. Server eindeutig identifiziert werden.

Ein große Vorteil ist, dass man, ohne ein System kompromittieren zu müssen, die interne Netzwerkstruktur eines beliebigen Netzwerkes eruieren kann – vorausgesetzt, man schafft es, den JavaScript Code auf einem beliebigen Rechner des Netzwerkes ausführen zu lassen.

Ein Nachteil ist die lange Dauer der Prozedur. Da die Tests timeout-basiert sind, liegt die Dauer davon im Sekundenbereich, was für effektive Tests zu lange ist. [SPI-06/1]

7.2.4.2. JavaScript + CSS

Prinzipiell ist es nicht möglich, direkt auf den Verlauf eines Browsers zuzugreifen.

Mit Hilfe von CSS können jedoch Links auf kürzlich besuchte Webseiten anders formatiert werden als Links zu nicht besuchten Seiten. Durch JavaScript können dynamisch Hyperlinks generiert werden, welche vom Browser automatisch gerendert werden. Die Formatierung dieser Links kann mittels JavaScript ausgelesen werden. Da dies in sehr kurzer Zeit möglich

ist, lassen sich mit dieser Technik sehr viele Links in kurzer Zeit überprüfen und damit die Information eruieren, ob bestimmte Webseiten kürzlich besucht wurden oder nicht.

Eine denkbare Anwendung für diese Kombination von JavaScript mit Cascading Style Sheets ist die Überprüfung bestimmter Suchworte in Google.

Wird ein Suchwort in Google abgesetzt, so ist die daraus resultierende URL der Ergebnisseite natürlich immer gleich. So ist es möglich, bestimmte Suchworte mit Hilfe einer Liste zu überprüfen.

Das gleiche Prinzip lässt sich natürlich auf alle Webseiten anwenden, welche im Verlauf gespeicherte, eindeutige URLs generieren.[SPI-06/2]

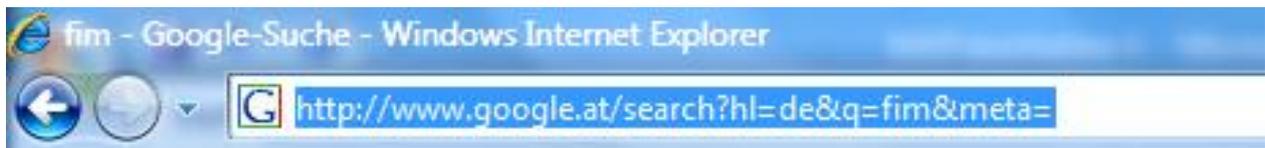


Abbildung 27: URL einer Google-Suche

7.3. Integration in *security.fim.uni-linz.ac.at*

Die vorgestellten Techniken zur Informationssammlung werden zweifach in *security.fim.uni-linz.ac.at* eingebunden. Einerseits als Widget, welches dem Benutzer die gesammelten Informationen in gekürzter Form präsentiert. Andererseits auch als eigenes Plugin, das in eine Unterseite von „Sicherheit“ eingebettet wird und die verwendeten Techniken genauer erläutert bzw. auch Möglichkeiten des Schutzes vorstellt.

8. Fazit und Ausblick

Das Security-Portal soll beides sein – sowohl Werbe- und Informationsträger für das Studium „Netzwerke und Sicherheit“ als auch Anlaufstelle für alle fachlich Interessierten. Die vorliegende Implementierung wird diesem Anspruch gerecht. Sowohl Informationen zum Studium als auch fachspezifische Informationen werden geboten, wobei vor allem die Quantität an weiterführenden und detaillierteren Informationen in Form von wissenschaftlichen Artikeln und Studentendarbeiten erst im Laufe der Zeit steigen wird.

Typo3 als Content Management System zu benutzen war sicherlich die beste, wenn auch nicht die einfachste aller Möglichkeiten. Typo3 liegt in seiner Komplexität doch deutlich über den anderen getesteten Systemen. Vor allem die Einarbeitung in die vielen verschiedenen Konzepte, die oft auf verschiedenen Wegen gleiche Resultate liefern (man denke hier z.B. an

die Lokalisierungsmöglichkeiten) nimmt viel Zeit in Anspruch. Oft stellt auch die Kombination von verschiedenen Konzepten eine Herausforderung dar – etwa die Verbindung von JavaScripts mit dem eID-Ajax-Konzept und der Lokalisierung von Komponenten.

Die hier vorgestellten Features und Inhalte des Portals stellen eine Momentaufnahme dar. Das Portal ist nun nicht fertig – ganz im Gegenteil:

Die in dieser Arbeit vorgestellte und erarbeitete Webseite stellt ein Gerüst, einen Ausgangspunkt dar. Ein Webportal wie *security.fim.uni-linz.ac.at* lebt von der Veränderung. Von der kontinuierlichen Anpassung und Ergänzung der Inhalte, aber natürlich auch von technischen und strukturellen Verbesserungen.

Das Portal wird hoffentlich Studierende wie Lehrende in ihren Zielen und Aufgaben unterstützen.

Dem Team von Netzwerke und Sicherheit viel Freude auf und mit *security.fim.uni-linz.ac.at!*

9. Literatur

- [BRO-08] Dokumentation und Quellcode der vielseitigen JavaScript-Boxen, der „Docking Boxes“, auf den Seiten von Brothercake. <http://www.brothercake.com/site/resources/scripts/dbx/>
- [BUY-07] Präsentation zum Status von Drupal und zukünftigen Entwicklungen, <http://www.scribd.com/doc/950328/State-of-Drupal-presentation-September-2007>
- [CMS-07] Webseite, spezialisiert auf den Vergleich von Content Management Systemen, www.cmsmatrix.org
- [DAM-06] Karsten Dambekalns, 2006. Typo3 4.0: DBAL. Datenbankzugriff mal anders, T3N Magazin für Typo3 und OpenSource. Quelle: http://www.yeebase.com/fileadmin/t3n/archiv/05-02/t3n_05-02_typo3_40_dbal.pdf
- [DRC-07] Informationen zur Menüerstellung in Drupal, <http://www.drupalcenter.de/handbuch/6082>
- [DRC-08] Referenzen von Drupal, <http://www.drupalcenter.de/handbuch/referenzen>
- [DRU-08/1] Systemanforderungen von Drupal, <http://drupal.org/requirements>
- [DRU-08/2] General concepts in Drupal, <http://drupal.org/node/19828>
- [DRU-08/3] Theme guide (Drupal 6), <http://drupal.org/theme-guide>
- [DRU-08/4] Informationen zur Sicherheit in Drupal, <http://drupal.org/security>
- [DRU-08/5] Seiten des Security-Teams von Drupal, <http://drupal.org/security-team>
- [EBE-07] Anja Ebersbach, Markus Glaser, Radovan Kubani. Joomla! Das Handbuch für Einsteiger. Galileo Computing. Online-Ausgabe des Buches, auf <http://www.galileocomputing.de/openbook/joomla/>
- [GRA-07] Hagen Graf, Joomla! 1.5, Websites organisieren und gestalten mit dem OpenSource-CMS, Addison-Wesley. Online-Ausgabe des Buches auf <http://joomla.cocoate.com/joomla-15>
- [JOO-08/1] Blogeintrag über das Entwicklerteam von Joomla! auf der offiziellen Seite von Joomla!, <http://www.joomla.org/content/blogcategory/13/29/>
- [JOO-08/2] Artikel über die minimalen Anforderungen von Joomla! auf der offiziellen Seite von Joomla!, <http://help.joomla.org/content/view/1938/310/>

- [JOO-08/3] Artikel über die Development Working Group von Joomla!,
<http://dev.joomla.org/content/view/1992/53/>
- [JOO-08/4] Kurzinformatio zu Sicherheitsangelegenheiten in Joomla!,
<http://dev.joomla.org/content/view/14/55/>
- [PÖT-06] Daniel Pöttinger, Blogeintrag zu Mehrsprachigkeit und TemplaVoila,
<http://www.typo3-media.com/blog/article/mehrsprachigkeit-und-templavoila.html>
- [SKÅ-04/1] Kasper Skårhøj, Modern Template Building, Part 1, 2004. Tutorial. Quelle:
http://typo3.org/documentation/document-library/tutorials/doc_tut_templselect/current/
- [SKÅ-04/2] Kasper Skårhøj, Robert Lembke, Futuristic Template Building, Tutorial. Quelle:
http://typo3.org/documentation/document-library/tutorials/doc_tut_ftb1/current/
- [SKÅ-04/3] Kasper Skårhøj, Getting Started, Tutorial, 2003. Quelle:
http://typo3.org/documentation/document-library/tutorials/doc_tut_quickstart/current/
- [SKÅ-06/1] Kasper Skårhøj, Frontend Localization Guide, Beschreibung der Konzepte zur Mehrsprachigkeit in Typo3, http://typo3.org/documentation/document-library/core-documentation/doc_l10nguide/current/
- [SKÅ-07/1] Kasper Skårhøj, TypeScript Syntax and In-Depth Study. Quelle:
http://typo3.org/documentation/document-library/core-documentation/doc_core_ts/current/
- [SKÅ-07/2] Kasper Skårhøj, Inside Typo3, Einführung in Konzepte von Typo3, Quelle:
http://typo3.org/documentation/document-library/core-documentation/doc_core_inside/current/
- [SKÅ-07/3] Kasper Skårhøj, Typo3 Core API, Beschreibung der Core API von Typo3 im Hinblick auf die Entwicklung von Extensions,
http://typo3.org/documentation/document-library/core-documentation/doc_core_api/4.1.0/view/
- [SPI-06/1] SPI Labs. Detecting, Analyzing, and Exploiting Intranet Applications using JavaScript, 2006
- [SPI-06/2] SPI Labs. Stealing Search Engine Queries with JavaScript, 2006

- [TVK-08] Artikel über die Geschichte von Joomla! auf einer Webseite, die verschiedene Informationen und Dienste rund um Joomla! anbietet. Quelle: <http://clubtvk.com/content/view/42/50/>
- [TYP-07/1] Artikel zur Frage „Was ist ein CMS?“ auf typo3.com http://typo3.com/What_is_a_CMS.1351.0.html
- [TYP-07/2] Artikel über die Geschichte von Typo3 auf typo3.com <http://typo3.com/History.1268.0.html>
- [TYP-07/3] Artikel über die an der Entwicklung/Wartung von Typo3 beteiligten Teams <http://typo3.org/teams/>
- [TYP-07/4] Seiten des Security-Teams von Typo3, <http://typo3.org/teams/security/>
- [TYP-07/5] Fakten über Typo3, http://www.typo3.com/Facts_and_Figures.factsandfigures.0.html
- [WIK-07] Artikel in der Wikipedia über Content Management Systeme, http://de.wikipedia.org/wiki/Content_management_system

Alle Internetlinks wurden am 16. Oktober 2008 auf ihre Gültigkeit überprüft.

Karl Thier, Bakk.techn.

geboren am 24. März 1985 in Melk
Landfriedstetten 16, 3252 Petzenkirchen
Tel.: 0664/4080506
e-mail: karljun.thier@aon.at



Ausbildung/Studium

1991 - 1995	Volksschule Petzenkirchen
1995 - 2003	BG/BRG Wieselburg Schüleraustausch mit der Österreichischen Schule in Budapest Matura mit Auszeichnung
2003 - 2007	Bachelor-Studium der Informatik an der Johannes Kepler Universität Linz
seit 2007	Master-Studium der Informatik an der JKU Linz

Berufliche Tätigkeit

2002 - 2005	Diverse Ferialpraktika im Bereich IT/Applikationsentwicklung
09/2007 - 12/2007	Wissenschaftlicher Mitarbeiter ohne Diplom für Forschung am Institut für Informationsverarbeitung und Mikroprozessortechnik an der JKU Linz
01/2008 - 03/2008	Mitarbeit bei der Organisation und Durchführung des Wahlkampfs für BR Bettina Rausch im Rahmen der Landtagswahl in NÖ

Außerschulische Weiterbildung/Aktivitäten

2002 - 2004	Teilnahme an den IX., X. und XI. Wachauer Journalistentagen
2004	Teilnahme an der European Union Simulation mit Prof. Reinhard Heinisch im Rahmen von Project Next 1
seit 2006	Teilnahme am Project Next 2
2006	Mitarbeit beim NR-Wahlkampf für DI Günther Hützl
seit 2007	Kooptiertes Mitglied im Gemeindeparteivorstand der ÖVP Bergland
seit 2007	Kommunikationsreferent im Vorstand der JVP Bezirk Melk

Fremdsprachen

Englisch - 8 Jahre AHS
Französisch - 4 Jahre AHS Oberstufe

Interessen

Lesen - politische Literatur, Zeitgeschichte
Philosophie, Diskussionen, Freunde treffen, gutes Essen
Laufen, Spazieren gehen, Ausdauer- und Krafttraining

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplom- bzw. Magisterarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Karl Thier