Konzeption und Entwicklung eines modularen Content Management Systems für gruppenbasiertes Bearbeiten

Bachelor-Thesis

zur Erlangung des akademischen Grades B.Sc.

Keno Weßels

2087166

Hochschule für Angewandte Wissenschaften Hamburg
Fakultät Design, Medien und Information
Department Medientechnik

Erstprüfer: Prof. Dr. Edmund Weitz

Zweitprüfer: Prof. Dr. Andreas Plaß

Hamburg, 23. 07. 2015

Inhaltsverzeichnis

Abstract

This thesis deals with the development of a new content management system (CMS) for the company Karnbrock GmbH. The aim was to replace an existing CMS and improve it in several key points. The new system was supposed to be designed as a modular CMS for group-based editing. This study documents the development process of this new CMS called "Synfo-2015". After outlining the purpose and the motivation for developing the new system in chapter 1, chapter 2 analyses the requirements for Synfo-2015 based on an assessment of the existing System. Chapter 3 then presents and explains the technologies that were chosen as the most suitable to meet these requirements. Chapter 4 explains how the required modularity was implemented in Synfo-2015. To this end, the basic structure of the system was divided into the three areas Synfo-skeleton, Synfo-core and Synfo-modules. Subsequently, the chapter addresses group-based editing as the second core requirement for Synfo-2015 and illustrates how the implementation of a flexible group-based rights system allows to accommodate individually customised hierarchical group structures adapted to the needs of the intended customers. The conclusion states that Synfo-2015, representing the result of half a year of development, provides a solid basis for the commercial use as a modular CMS for group-based editing for the company Karnbrock GmbH.

Zusammenfassung

Diese Arbeit befasst sich mit der Neuentwicklung eines Content Mangement Systems (CMS) für die Firma Karnbrock GmbH. Ziel der Neuentwicklung war es, ein vorhandenes CMS zu ersetzten und in zentralen Punkten zu verbessern. Das neue System “Synfo-2015“ sollte als modulares CMS für gruppenbasiertes Bearbeiten angelegt sein. Die vorliegende Arbeit dokumentiert diesen Entwicklungsprozess. Dazu werden zunächst anhand der Beschreibung des bestehenden Systems die Anforderungen für Synfo-2015 herausgearbeitet (Kapitel 2). Daraufhin wird dargelegt und begründet, welche Technologien für die Umsetzung dieser Anforderungen an Synfo-2015 als am besten geeignet erachtet wurden (Kapitel 3). Kapitel 4 erläutert, wie die angestrebte Modularität in Synfo-2015 umgesetzt wurde. Dafür wurde in der Grundstruktur des Systems eine Aufteilung in die drei Bereiche Synfo-Skelett, Synfo-Kern und Synfo-Module vorgenommen. Anschließend befasst sich die Arbeit mit der zweiten Kernanforderung an Synfo-2015 und stellt dar, wie ein flexibles gruppenbasiertes Rechtesystem implementiert wurde, das es erlaubt, hierarchische Gruppenstrukturen von den für das CMS vorgesehenen Kunden abzubilden. Im Fazit wird festgestellt, dass nach einem einem halben Jahr Entwicklung ein Ergebnis vorliegt, das eine solide Grundlage für ein auf die kommerzielle Nutzung ausgelegtes modulares CMS für gruppenbasiertes Bearbeiten für die Firma Karnbrock GmbH bietet.

1 Einleitung

Die Karnbrock GmbH wurde als Internet-Agentur im Jahr 2004 gegründet und wächst seitdem stetig. Sie besteht im Jahr 2015 aus insgesamt sieben Mitarbeitern und bietet angepasste Internetauftritte für mittelständische Unternehmen an. Das breite Spektrum an Kunden reicht von einem Ökostromanbieter über Theater und andere Kultureinrichtungen bis hin zu großen deutschen Stiftungen[1]. Im Laufe der letzten Jahre hat sich durch zahlreiche Aufträge von verschiedenen universitären Instituten, unter anderem der Universität Bremen, ein zusätzlicher Kundenstamm entwickelt.

Da sich die umgesetzten Internetseiten für die verschiedenen wissenschaftlichen Institute sehr ähneln, wurde die Idee zu einem Produkt entwickelt, das später die Bezeichnung “Synfo“ erhielt. Bislang verkauft die Firma Karnbrock GmbH nur Internet- und Intranet-Lösungen auf Auftragsbasis.

Das dreigeteilte System wurde zuerst im Jahr 2012 für das Institut “Zentrum für Sozialpolitik“ der Uni Bremen umgesetzt. Bei diesem und weiteren Kunden kommt es so gut an, dass das Produkt Synfo seit Ende des Jahres 2014 eine eingetragene Marke der Firma Karnbrock GmbH ist und als solches beworben wird. In der Firmengeschichte ist es das erste Mal, dass ein eigenständiges Produkt entwickelt und vertrieben werden soll.

Die Internetseiten, die für die Institute umgesetzt wurden, verfügen alle über ein speziell für diese Einrichtungen angepasstes Content Mangement Systems (CMS). Dabei haben die verschiedenen Mitarbeiter jeweils individuellen Zugang und können Inhalte, für die sie Administrationsrechte haben, auf der Webseite bearbeiten. Außerdem können sie ihre Publikationen, Veranstaltungen, Projekte und Neuigkeiten verwalten. Zusätzlich zu den Zugängen der Mitarbeiter gibt es separate Zugänge für die Abteilungen und für das Institut als höchster Ebene.

[1] Karnbrock GmbH, *Karnbrock GmbH Referenzen*

Dieses System besteht aus einem über viele Jahre gewachsenen Quellcode. Dieser ist weder versioniert, noch hat er einen einheitlichen, systematischen Entwicklungsstrang. So kann Synfo zwar verkauft werden, das System ist allerdings weder gut skalierbar noch auf modernen Technologien aufgebaut. Für die Karnbrock GmbH ergibt sich daraus, dass zum einen für die Erstellung jedes neuen Projekts für einen neuen Kunden sehr viel Zeit und Ressourcen aufgewandt werden mussten. Zum anderen war der spätere Wartungsaufwand nach der Online-Stellung groß. Deshalb war es notwendig, das System neu zu entwickeln oder zu modernisieren, wozu sich die Firma aber auch aus finanziellen Gründen noch nicht entschlossen hatte.

Zu diesem Zeitpunkt arbeitete ich bereits über zwei Jahre als Werkstudent bei der Karnbrock GmbH. Da ich die Idee, ein komplexes System zu entwerfen und zu entwickeln, sehr spannend fand, schlug ich vor, dass ich mich mit der Problematik befassen und in meiner Bachelorarbeit das System neu konzipieren und entwickeln könnte. Dieser Vorschlag wurde von dem Firmeninhaber und den Kollegen begrüßt.

1.1 Zielsetzung

Das Produkt Synfo – hier im Weiteren als Synfo-2012 bezeichnet – gibt es noch nicht als konsistentes System. Um es zukunftsfähig zu machen, müsste der Quellcode, der aktuell in verschiedenen Projekten existiert und sich in unterschiedliche Richtungen weiter entwickelt hat, entweder zusammengefasst, umgeschrieben und verallgemeinert oder komplett neu geschrieben werden.

Es erschien sinnvoller, ein neues System zu entwickeln, das zwar an Synfo-2012 anschließt, aber von Beginn an neu aufgesetzt ist. Das neue, im Folgenden als Synfo-2015 bezeichnete System soll verschiedenen Anforderungen gerecht werden. Zum einen sollen zunächst grundsätzliche konzeptionelle Fehler, die zu Beginn von Synfo-2012 gemacht wurden, vermieden werden können. Zum anderen sollte das System skalierbar sein, damit verschiedene Kunden auch verschiedene Einstellungsmöglichkeiten haben, wie zum Beispiel verschiedene Sprachen, Designs, Funktionen und vieles mehr. Gleichzeitig sollte es modular sein, damit es jedem Kunden ermöglicht, individuell und flexibel Module zu installieren oder wegzulassen.

1.2 Struktur dieser Arbeit

An der Entwicklung eines an die Bedürfnisse der Firmenkunden angepassten Content Management Systems für gruppenbasiertes Bearbeiten habe ich ein halbes Jahr in der Firma Karnbrock GmbH gearbeitet. Das Ergebnis am Ende dieser praktischen Phase ist ein modulares Grundgerüst mit definierter Rechtevergabe als Grundlage für die Zusammenstellung aller benötigten Funktionen. Eine erste, auf dem neu entwickelten Synfo-2015-System basierende Internetseite soll ein Jahr nach Start des Projektes, im November 2015, online gehen. Dann soll Synfo-2015 die ältere Version ablösen.

Da diese Arbeit einen praktischen Arbeitsprozess begleitet und dokumentiert, ist sie auch entsprechend den Schritten dieses Arbeitsprozesses strukturiert. Am Anfang steht die Analyse des aktuellen Standes und im zweiten Schritt folgt die Ausarbeitung einer spezifisch angepassten Zielsetzung. Auf der Grundlage der Sichtung und begründeten Auswahl der zur Verfügung stehenden Technologien basieren die nächsten Schritte, die Konzeption und die Implementierung. Wobei die Schwerpunkte bei dem modularen Aufbau und die dafür zu entwickelnde Schnittstelle sowie dem flexiblen Gruppensystem liegen. Dabei wird darauf verzichtet, die gesamte Implementierung genau zu beschreiben, da allein der Synfo-Kern fast 15.000 Zeilen beinhaltet. Es werden jedoch zur Veranschaulichung des Arbeitsprozesses zu einigen Funktionalitäten beispielhaft Quellcodeausschnitte präsentiert. Am Schluss steht ein Fazit einschließlich einer Bewertung.

2 Analyse

2.1 Istzustand

Das System Synfo-2012 stellt die Basis für die Entwicklung von Synfo-2015 dar. Als erste Version war es angelegt als ein erweitertes Content Management System (CMS) zum Erstellen, Bearbeiten und Verwalten von Websites. Außerdem ermöglicht es Nutzern, Veranstaltungen, Neuigkeiten, Publikationen und Projekte zu verwalten. Es gibt drei verschiedene Login-Bereiche für die verschiedenen Ebenen innerhalb eines wissenschaftlichen Instituts. Jede dieser Ebenen hat unterschiedliche Rechte und Zugriffsrechte. Auf der untersten Ebene kann der einfache Mitarbeiter seinen persönlichen Bereich auf der Webseite verwalten und eigene Seiten und Unterseiten mit verschiedenen Templates erstellen. Die nächste Ebene ist den verschiedenen Abteilungen innerhalb des Instituts zugeordnet. Die oberste Ebene ist der Institutsleitung vorbehalten. Dort werden die Inhalte und Seiten verwaltet, die zu der Einrichtung gehören. Auf allen Ebenen können Publikationen und andere Inhalte wie Neuigkeiten, Veranstaltungen oder Projekte erstellt und bearbeitet werden. Neu erstellte Inhalte können jeweils der übergeordneten Ebene vorgeschlagen werden. Wenn also ein Mitarbeiter zum Beispiel eine Veranstaltung oder andere Inhalte innerhalb seiner Abteilung erstellt, so kann er der darüber liegenden Ebene, der Institutsleitung, vorschlagen, diese Inhalte auch in ihren Bereich zu übernehmen. Die Institutsleitung, kann dann diese Inhalte annehmen, ablehnen oder anpassen und annehmen. Wenn die Inhalte akzeptiert werden, werden sie auch innerhalb ihrer Ebene auf der Internetseite angezeigt. Jeder Mitarbeiter, der in mehreren dieser Ebenen tätig ist, braucht bei Synfo-2012 allerdings für jeden Bereich einen eigenen Benutzernamen und ein gesondertes Passwort, und er muss sich, um die Ebenen zu wechseln, aus- und wieder einloggen.

Es wurden bereits fünf dieser bestehenden CMS Lösungen an verschiedene Kunden verkauft, welche alle im Kern dieselben Funktionen erfüllen. Dennoch mussten sie jedes Mal aufwändig für den jeweiligen Kunden angepasst werden. Diese alten Systeme weisen also zwar viele Ähnlichkeiten auf, es gibt aber keinen definierten Kern, der bei allen Lösungen identisch ist, was den Arbeitsaufwand für die Firma Karnbrock GmbH erheblich verringern und dadurch die Wirtschaftlichkeit erhöhen würde. Diese CMS-Systeme (Synfo-2012) wurden und werden zum Teil parallel entwickelt. Ihr Funktionsumfang deckt bereits sehr viele Bedürfnisse der Kunden ab. Im Prinzip bietet Synfo-2012 aus der Sicht des Kunden die meisten Funktionalitäten, die auch das neue Synfo-2015 definiert. Da das alte System aber nicht versioniert ist, ist es für die Firma Karnbrock GmbH schwierig, den Überblick zu behalten, an welcher Stelle für welchen Kunden welches Feature umgesetzt wurde. Bei der Entwicklung wurden außerdem keine definierten Konventionen eingehalten, weshalb die genutzte Technologie teilweise veraltet und der Quellcode für Externe nur schwer verständlich ist. Zusätzlich ist das alte System nicht skalierbar. Es gibt keine Möglichkeit, in wenigen Schritten bestimmte, später als Module beschriebene Funktionen und Features hinzuzufügen, geschweige denn zu konfigurieren. Alle Sicherheitsupdates müssen für jeden Kunden einzeln umgesetzt werden, da es keinen gemeinsamen definierten Kern gibt. Zwar funktioniert das bestehende System, und es hat die für den Kunden wichtigsten Funktionen implementiert. Jedoch ist absehbar, dass der Support-Aufwand für das System mit jeder weiteren verkauften Lösung exponentiell wachsen wird, weil es kein definiertes, geschlossenes System ist. Außerdem ist der Aufwand für jede neue Installation groß, da jedes einzelne System einmal aus einer Vorgängerinstallation dupliziert und neu angepasst werden muss.

2.2 Anforderung

Aufgrund dieser Missstände ergeben sich verschiedene Anforderungen an ein neues System, die im Folgenden beschrieben werden.

Es sollte ein Grundsystem “Webseite mit CMS“ geben und dazu weitere “Module“, welche von dem Kunden unabhängig voneinander zu dem Grundsystem hinzu gebucht werden können. Diese Module sollten jeweils für sich funktionieren, und es sollte möglich sein, sie ohne großen Aufwand anzukoppeln, zu entfernen oder auszutauschen,

das heißt, sie sollten modular sein. Diese einzelnen Module sollten auch untereinander verknüpft werden können, ohne dass sie voneinander abhängig sind.

Durch diese Standardisierung bei gleichzeitiger Variabilität werden weitere Anforderungen definiert: Die einzelnen Module sollten für jeden Kunden gleich sein. Lediglich die Konfiguration dieser Module sollte sich von Kunde zu Kunde unterscheiden dürfen. So sollte erreicht werden, dass diese einzelnen Module für alle Kunden gleichzeitig weiter entwickelt werden können. Daraus ergibt sich im nächsten Schritt, dass die Module versioniert sein sollten und somit Versionsnummern erhalten, wodurch definiert werden kann, welches Modul mit welchem in welcher Version kompatibel ist. Für die einzelnen Kunden sollten also im bestmöglichen Fall vom Anbieter nur Konfigurationsdateien angepasst und individuelle Layouts erstellt werden müssen.

Eine weitere wichtige Anforderung an die Neuentwicklung ist, dass das Produkt als solches stetig wachsen und erweitert werden kann. So sollte zum einen gesichert werden, dass der Kern mit weiteren Funktionen oder Sicherheitsupdates angepasst werden kann. Zum anderen sollte es ermöglicht werden, dass jedes Modul einzeln wachsen und jedes für sich in neuen Versionen "erscheinen" kann. Davon profitiert letztlich auch der Kunde, der neue Funktionen, die im Laufe der Zeit zu den entsprechenden Modulen hinzugefügt werden, erhalten und so immer auf dem neuesten Stand bleiben kann. Außerdem profitierten die Entwickler, die nicht jedes Sicherheitsupdate oder jede Funktion in vielen verschiedenen Systemen parallel einbauen müssen.

Das System sollte, so die Anforderung des Anbieters, ein komplexes Benutzersystem haben. Es sollte möglich sein, viele Benutzer in verschiedenen Gruppen mit verschiedenen Rechten abzubilden. Die Aufspaltung der Nutzer durch drei verschiedene Login-Bereiche in Synfo-2012 ist nicht benutzerfreundlich. Deshalb sollte Synfo-2015 mit nur einem Login-Bereich auskommen. So können Benutzer, die in verschiedenen Ebenen innerhalb des Instituts tätig sind, alles verwalten, ohne sich für einen Bereichswechsel aus- und wieder einloggen zu müssen.

Aus den hier definierten Anforderungen ergibt sich ein relativ komplexes Bild, wie das System aufgebaut sein sollte. Um dem gerecht zu werden, gibt es verschiedene Möglichkeiten: Entweder kann das System mit einem schon auf dem Markt befindlichen CMS umgesetzt werden, wobei dieses um die fehlenden Funktionen erweitert werden müsste. Die andere Möglichkeit ist die eigenständige Entwicklung eines kompletten

CMS mit der Anpassung an die besonderen Anforderungen der Kunden der Firma Karnbrock GmbH.

Ein bereits auf dem Markt befindliches CMS könnte entweder Open Source sein und ohne Kosten verwendet oder eingekauft werden. Ein CMS, das die wichtigsten Anforderungen bereits erfüllt, wurde nicht gefunden. Gerade die sehr komplexe und angepasste Benutzerverwaltung, welche das Kernelement von Synfo-2015 bilden soll, wurde in keinem untersuchten CMS auf Anhieb gefunden. Da die Benutzerverwaltung allerdings Kernelement der meisten CMS ist, wäre ein Eingriff in das genutzte CMS unvermeidbar gewesen. Ein solcher Eingriff hätte damit die Möglichkeit, das CMS zu aktualisieren, erheblich gemindert oder besonders aufwändig gemacht. Aufgrund der sehr speziellen Benutzerverwaltung, die für Synfo-2015 notwendig ist, wurde also von der Benutzung eines schon bestehenden CMS abgesehen.

Obwohl das Erstellen eines komplett neuen CMS viel Aufwand bedeutet, sprach vieles dafür, ein eigenes System zu entwickeln. Der Vorteil eines komplett neu entwickelten CMS besteht darin, dass das System nur genau das abbildet, was es tatsächlich benötigt. Alle Bereiche des CMS können darauf angepasst werden, dass es verschiedene Gruppen gibt und dass jeder Benutzer in verschiedenen Gruppen verschiedene Rollen vertreten kann.

Zusammenfassend kann also festgehalten werden:

- Mit Synfo-2015 soll ein Produkt entwickelt werden, dass von der Funktionalität so aufgebaut ist wie Synfo-2012 (differenzierte Zugänge zum Verwaltungsbereich, hierarchisches Gruppensystem und Inhaltsbearbeitung der Bereiche News, Publikationen etc.). Dabei sollen Fehler vermieden werden, die bei der Konzeption und Umsetzung von Synfo-2012 gemacht wurden.
- Dadurch, dass Synfo-2015 als eigenständiges Produkt entwickelt und vertrieben werden soll, kann es ein konsistenteres System abbilden.
- Synfo-2015 soll als Produkt für verschiedene Kunden einsetzbar und trotzdem als einheitliches System pflegbar sein.
- Es soll auf aktuelle Technologien gesetzt werden, um die Zukunft des Produktes zu sichern.

- Durch eine modulare Bauweise soll das Produkt stetig erweiterbar und anpassbar sein.
- Module, die für einzelne Kunden entwickelt werden, sollen auch für weitere Kunden ohne Aufwand einsetzbar sein.
- Die Benutzerverwaltung soll skalierbarer für verschiedene Kunden und mit unterschiedlichen Anforderungen flexibel einsetzbar sein.

3 Technologiewahl

Für die Umsetzung von Synfo-2015 werden verschiedene Technologien eingesetzt. Zunächst muss festgelegt werden, mit welchen Programmiersprachen das System umgesetzt werden soll. Auf die Programmiersprache kann gegebenenfalls ein Framework aufgesetzt werden, das bei vielen Aktionen behilflich sein kann. Des weiteren werden für einen funktionierenden Workflow und die Modularität des Systems die Technologien Composer und Vagrant eingesetzt.

3.1 Die Programmiersprache PHP

Für die Umsetzung eines CMS im Web stehen verschiedene Programmiersprachen zur Verfügung. Je nach Anwendungsgebiet und Komplexität des Systems bieten sich Java, ASP.NET, Ruby, PHP sowie serverseitiges Javascript (Node.js) an. Für die Entscheidung sind nicht nur qualitative oder formale Faktoren von Bedeutung, sondern auch der Rahmen, in dem das System umgesetzt werden muss. Da keiner der Mitarbeiter der Firma Karnbrock GmbH nennenswerte Kenntnisse in den Programmiersprachen ASP.NET, Ruby oder Java hat, werden diese von vorn herein nicht in Betracht gezogen. Als Programmiersprachen kommen also nur PHP und Node.js in Frage. Ein CMS auf Grundlage von Node.js ist denkbar. Mit Node.js ist es möglich, Javascript Quellcode serverseitig auszuführen. Dadurch lassen sich Webseiten mit Javascript entwickeln. Node.js ist eine neue Technologie aus dem Jahr 2009[2], mit der die Firma Karnbrock GmbH bislang noch wenig Erfahrung gemacht hat. Da für die Entwicklung von Synfo-2015 zu einem späteren Zeitpunk eventuell auf Freelancer zurückgegriffen werden soll, muss sichergestellt sein, dass es genug Freelancer gibt, die die gewählte Programmiersprache beherrschen. Durch die Eingabe verschiedener Suchbegriffe für Freelancer auf

[2]Teixeira, 2012, S. 3.

der Internetplatform Gulp wurde geprüft, wie häufig Programmiersprachen von Freelancern angeboten werden. Gulp ist eine Internetplattform zur Vermittlung zwischen Firmen und Freelancern. Abbildung 3.1 zeigt die Anzahl der Suchergebnisse nach bestimmten Suchbegriffen. Die Ergebnisse dieser Suchen verdeutlichen, dass es zwar viele Javascript-Entwickler gibt. Allerdings haben wenige davon Node bzw. Node.js als Kompetenz angegeben, woraus man schließen kann, dass diejenigen, die Javascript als Kompetenz angeben, aus dem Frontend-Bereich kommen und kaum Kentnisse bei der Umsetzung von Backends mit Javascript haben. Für PHP und auch ASP.NET gibt es im Vergleich dazu viele Freelancer.

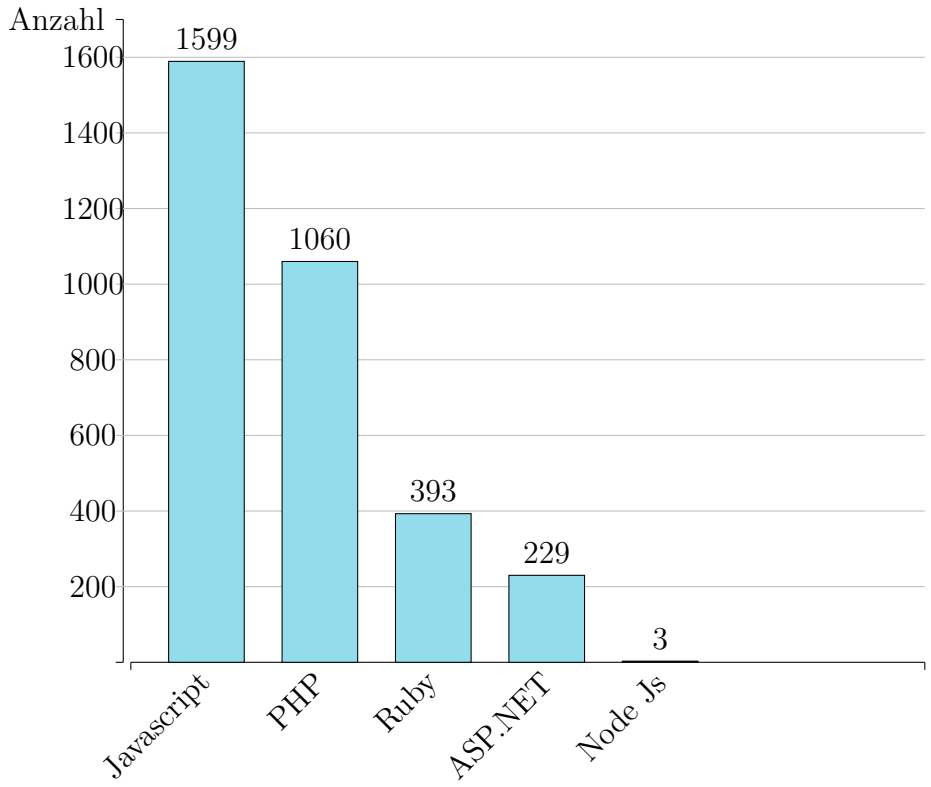

Abbildung 3.1: Anzahl Suchergebnisse Freelancer auf http://www.gulp.de[3] vom 19.06.2015

Darüber hinaus arbeitet die Firma Karnbrock GmbH seit jeher nur mit PHP für die Umsetzung ihrer Internetlösungen, so dass das Team der Karnbrock GmbH weniger Kenntnisse in anderen Programmiersprachen im Bezug auf Internetseiten hat.

[3]GULP Information Services GmbH, *Gulp - Homepage*

PHP ist eine relativ alte Sprache, die bereits im Jahr 1995[4] in einer ersten Version erschienen ist. Sie wird allerdings nach wie vor für viele Internetseiten verwendet. Laut der Website W[3]techs, die Statistiken über Web Technologien bereitstellt, ist PHP mit 81,8% die meist genutzte serverseitige Programmiersprache für Internetseiten. Ihr folgen ASP.NET mit 16,09% und Java mit 3,0%, Stand 17.06.2015 [5].

PHP wird stetig weiterentwickelt. Im Jahr 2004 ist die erste Version von PHP 5 erschienen[6]. Seit der Version PHP 5 kann auch objektorientiert programmiert werden.

Es gibt viele Freelancer, die PHP-Kenntnisse haben. Auch ich selbst arbeite, zumindest im Web-Bereich, lieber mit PHP als mit den anderen Programmiersprachen, weil mir PHP gut geläufig ist und ich damit relativ schnell gute Ergebnisse erzielen kann. Deshalb fällt die Wahl auf PHP.

3.2 Das Framework Yii2

Ein Framework hilft in der Programmierung, Grundfunktionalitäten bereitzustellen, damit nicht jede Funktion oder Hilfsklasse für jedes Projekt einzeln entwickelt werden muss. Außerdem können Frameworks bei der Strukturierung des Projektes helfen. Frameworks bieten eine gute Grundlage, um darauf ein komplexes System zu entwickeln[7].

Synfo-2012 baut auf keinerlei Framework auf. Es ist komplett innerhalb der Firma Karnbrock GmbH mit PHP umgesetzt worden und über die Jahre immer weiter gewachsen. Es nutzt einige alte Libraries, die allerdings nicht kontinuierlich mit Updates gepflegt werden. Synfo-2015 soll hingegen möglichst von den Vorteilen eines Frameworks profitieren.

Für PHP gibt es aktuell viele Frameworks auf dem Markt. In der Firma Karnbrock GmbH wurden in den letzten Jahren einige Projekte mit dem Framework “Yii“ um-

[4]The PHP Group, *PHP Homepage - History of PHP*
[5]Q-Success DI Gelbmann GmbH, *w3techs - Usage of server-side programming languages for websites*
[6]The PHP Group, *PHP Homepage - History of PHP*
[7]Maurice, 2015, S. 447,448.

gesetzt, deshalb wurde in Betracht gezogen, auch die Neukonzeption von Synfo-2015 mit "Yii2" umzusetzen, das im Oktober 2014 erschienen ist[8]. Yii2 setzt, wie fast alle aktuellen PHP-Frameworks[9], auf das "Model View Controller"-Entwurfsmuster (MVC)[10]. MVC teilt den Quellcode in drei verschiedene Zwecke ein. Die Funktionsweise von MVC wird in Abbildung 3.2 gezeigt.

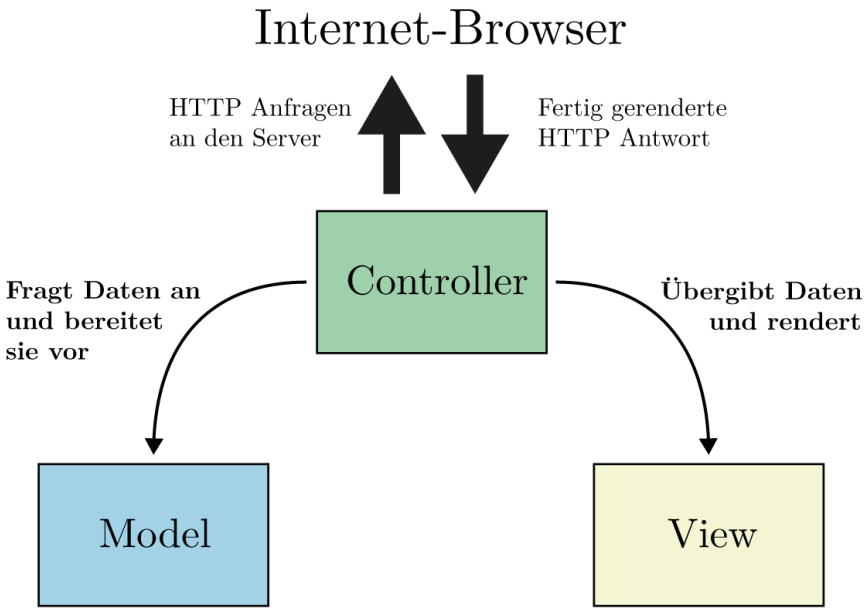

Abbildung 3.2: MVC im Web (Die Grafik ist einer Grafik von Meyen und Stropek[11] nachempfunden)

- **Model:** Die Models sind direkt mit den hinterlegten Daten verbunden. Bei Yii2 sind Models meist mit Datenbanktabellen verknüpft. Die Models sind dafür zuständig, die Inhalte und Daten bereit zu stellen. Auch bietet das Model Funktionen zum Verändern der Daten[12].
- **View:** In den Views befinden sich die Ausgabedateien. Views bestehen oftmals aus HTML-Dateien oder PHP-Dateien, ohne kritische Programmlogik zu enthalten. Views erhalten Inhalte von dem Controller.

[8] Yii Software LLC, *Yii 2.0.0 is released*
[9] Meyen und Stropek, 2013, S. 51.
[10] Yii Software LLC, *Yiiframework - Guide - How does Yii Compare with Other Frameworks?*
[11] Meyen und Stropek, 2013, S. 191.
[12] Geirhos, 2015, S. 521.

- **Controller:** Der Controller wird von dem Betrachter aufgerufen. Nachdem der Controller die Berechtigung für die Anfrage geprüft hat, lädt er, falls der Betrachter die Berechtigung hat, die angefragten Daten aus den Models und rendert die Views mit Hilfe der Daten. Die gerenderten Views werden an den Betrachter zurückgegeben.

Diese Einteilung des Quellcodes in verschiedene Aufgabenbereiche strukturiert den gesamten Quellcode. Außerdem können einzelne "Models" beispielsweise von verschiedenen "Views" auf unterschiedliche Art und Weise dargestellt werden, was den Quellcode weniger redundant macht. Darüber hinaus hat es den Vorteil, dass die gesamte Darstellung der Daten verändert werden kann, ohne dass Änderungen an der Logik vorgenommen werden müssen[13].

Zur Prüfung der Durchführbarkeit wurde Synfo-2015 zunächst testweise mit Yii2 umgesetzt. Da dies gut realisiert werden konnte, wurde schnell entschieden, weiter mit Yii2 zu arbeiten, ohne weitere Frameworks zu testen. Andernfalls wären alternativ die Frameworks Laravel und CakePHP in Frage gekommen.

3.3 Die Versionskontrolle mit Git und GitHub

Versionskontrolle beschreibt in der Programmierung und auch in der Textbearbeitung die "Verfolgung und Verwaltung von Änderungen"[14]. Grundlegend hilft die Versionskontrolle bei Projekten, an denen mehrere Entwickler parallel arbeiten, um nachvollziehen zu können, wer welche Änderungen vorgenommen hat und wann sie getätigt wurden[15]. In der Firma Karnbrock GmbH bekommt so im Rahmen der Versionskontrolle jedes Projekt ein eigenes Repository. Die Änderungen, die in der Vergangenheit an einem Projekt gemacht wurden, können jederzeit innerhalb eines Repositories nachvollzogen werden. Außerdem kann ein bestimmter Stand innerhalb dieser Änderungen einen "Tag" bekommen. Das bedeutet, dass die Stelle in der Versionierung markiert ist[16]. So kann eine ganz bestimmte Version definiert werden. Die Versionskontrolle ermöglicht also bei dem Projekt Synfo-2015, die Versionierung des Kerns

[13] Bruns und Meyer-Wegener, 2005, S. 255.
[14] Loeliger, 2010, S. 1.
[15] Ebd., S. 1.
[16] Ebd., S. 48.

und der einzelnen Module zu organisieren. Wie diese genau aufgebaut sind, wird in Kapitel 4.2 “Definition der modularen Struktur“ auf Seite 26 erläutert.

Üblicherweise sind SVN und Git die in der Praxis am meisten angewendeten Versionskontrollsysteme. In Abbildung 3.3 ist eine Statistik der Nutzung von SVN, Git und CVS der Jahre 2009 bis 2014 zu sehen. Die Eclipse Foundation macht seit 2009 jedes Jahr eine Umfrage. In dieser Umfrage geht es um verschiedene Aspekte in der Programmierung, unter anderem die Nutzung von Versionskontrollsystemen. Die Daten der Abbildung 3.3 basieren auf den Auswertungen dieser Umfragen[17]. Das Diagramm zeigt, dass Git im Laufe der letzten fünf Jahre stetig mehr Nutzer gewinnen konnte, während sowohl SVN als auch CVS mehr und mehr Nutzer verlieren. Git wird also von den Entwicklern immer häufiger eingesetzt und gewinnt in den letzten Jahren mehr an Bedeutung.

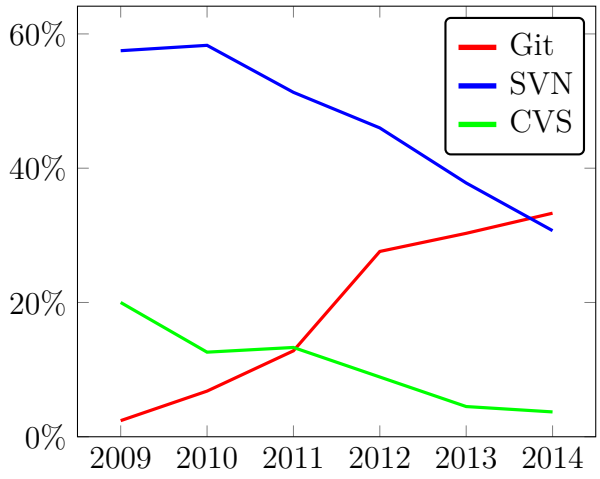

Abbildung 3.3: Nutzung von Git, SVN und CVS

Die Firma Karnbrock GmbH arbeitete bislang bei den meisten ihrer Projekte mit SVN. Allerdings hat Git viele Vorteile gegenüber SVN. Git ist darauf ausgelegt, in verschiedenen Branches, also parallelen Entwicklungssträngen, zu arbeiten, um verschiedene Versionen auch wieder zusammen fügen zu können[18]. Das ist bei SVN zwar auch möglich, aber nur mit relativ großem Aufwand umzusetzen. Git arbeitet im Gegensatz zu SVN dezentral, es muss daher keinen zentralen Server geben, auf dem die

[17] Eclipse Foundation, *eclipse survey 2009*; dies., *eclipse survey 2010*; dies., *eclipse survey 2011*; dies., *eclipse survey 2012*; dies., *eclipse survey 2013*; dies., *eclipse survey 2014*

[18] Git, *Git - Homepage - About*

Versionskontrolle läuft[19]. Bei Git werden Änderungen also erst lokal in einem "commit" gespeichert und können dann auf einen zentralen Server "gepusht" werden. Das ermöglicht auch das Versionieren ohne aktive Internetverbindung. Auch in der Firma Karnbrock GmbH wird bei wenigen Projekten bereits Git eingesetzt. Die Erfahrungen mit Git waren in diesen Fällen gut, wobei nur Entwickler beteiligt waren, die Interesse hatten, sich mit dem neuen Versionskontrollsystem auseinanderzusetzen. Unter diesen Voraussetzungen und da ich schon mehrere universitäre Projekte erfolgreich damit umgesetzt habe, wird bei Synfo-2015 mit Git gearbeitet.

Für Git ist es nicht zwingend notwendig, einen Hoster für das Repository zu haben, da Git auch dezentral funktioniert. Mit einem Hoster findet die Synchronisierung der Repositories aber an einer zentralen Stelle statt, und die Entwickler können alle auf diese zugreifen. Da das kollektive Arbeiten mit Git viel einfacher zu organisieren ist, wenn es einen zentralen Hoster gibt, soll Synfo-2015 auf einem Git-Hoster laufen.

Der aktuell von der Firma Karnbrock verwendete Hoster für SVN Repositories könnte auch zum Aufsetzen von Synfo-2015 in Frage kommen. Da Synfo-2015 ein größeres Projekt ist, das über die Zeit stetig wachsen soll, wird allerdings ein System benötigt, in dem Bugs, Verbesserungsvorschläge und Features gesammelt und organisiert, in "Milestones" aufgeteilt und Entwicklern zugewiesen werden können. Es gibt verschiedene Hoster, die zusätzlich zu dem einfachen Hosten eine Web-Plattform zur Verfügung stellen, mit der diese zusätzlichen Anforderungen abgedeckt werden können. Zu diesen Hostern gehören zum Beispiel "Bitbucket" und "GitHub"[20]. Diese beiden Hoster wurden zu Beginn der Entwicklung von Synfo-2015 daraufhin untersucht, ob sie alle Anforderungen von Synfo-2015 abdecken.

Bitbucket bietet für Privatkunden auch geschlossene Repositories an, wenn sie einen Gratis-Account nutzen. Bei GitHub sind geschlossene Repositories nur möglich, wenn der Account bezahlt ist. Für Organisationen muss bei beiden Plattformen gezahlt werden. GitHub gibt auf seiner Website an, der größte Hoster für Quellcodes zu sein[21]. Beide Hoster beinhalten eine Web-Oberfläche zum Verwalten von Repositories, Bugs, Issues, Milestones und Tags. Da in einem späteren Verlauf von Synfo-2015 durchaus

[19] Chacon und Straub, 2014, S. 29, 30.
[20] Fournova Software GmbH, *Tower - Homepage - 12 Git Hosting Services Compared*
[21] GitHub, Inc, *About Github*

die Möglichkeit besteht, externe Mitarbeiter für kleinere oder größere Module einzusetzen, ist es von Vorteil, ein System zu nutzen, in dem viele Entwickler bereits registriert sind und das allgemein als bekannt vorausgesetzt werden kann. Obwohl GitHub für Organisationen etwas teurer ist, überzeugen sowohl die Verwaltungsoberfläche als auch die großen Nutzerzahlen, weshalb GitHub für das Hosting von Synfo-2015 eingesetzt wird.

3.4 Weitere Technologien zur Hilfestellung der Projektentwicklung

In dem Projektverlauf wurde mit der Zeit klar, dass bestimmte weitere Technologien eingesetzt werden sollten. Im Folgenden wird auf diese eingegangen:

3.4.1 Composer als Dependency Manager

Composer wird als “Dependency Manager“ bezeichnet und grenzt sich von dem den Bekannten “Package Manager“ wie “Apt“ oder “npm“ ab. Der Unterschied zwischen einem “Package Manger“ und Composer als “Dependency Manager“ ist, dass Composer projektbezogen Abhängigkeiten installiert und nicht wie bei einem “Package Manager“ üblich, global für ein System[22]. So definiert Composer für PHP-Systeme, welche Add-Ons, Erweiterungen oder Libraries als “Pakete“ in welcher Version notwendig sind, damit das System funktioniert[23]. Für die Nutzung von Composer gibt es eine Einstellungsdatei im JSON-Format mit Informationen über das Projekt und mit der Definition, welche Pakete benötigt werden. Über den Konsolenbefehl “composer install“ werden alle vorher definierten Pakete heruntergeladen und dem Projekt hinzugefügt. Bei Synfo-2015 wird unter anderem mit Composer gearbeitet, weil Yii2 so einfacher zu aktualisieren und die Modularität bei PHP-Anwendungen einfacher umzusetzen ist. Das Framework Yii2 arbeitet vorzugsweise mit dem “Dependency

[22] *Composer - Introduction - Dependency management.*
[23] Adermann und Boggiano, ebd.

Manager“-Composer. Yii2 selbst wird im Normalfall installiert, indem es als “dependency“, also in Abhängigkeit zu einem Projekt hinzugefügt wird. In der Composer-Einstellungsdatei wird definiert, welches Modul geladen werden soll. So werden bei Synfo-2015 über Composer das Framework Yii2, der Snfo-Kern, die Synfo-Module sowie viele weitere benötigte Pakete geladen. Auf die Vorteile von Composer für die Modularisierung wird in Kapitel 4.2 “Definition der modularen Struktur“ auf Seite 26 weiter eingegangen.

3.4.2 Vagrant als Testumgebung

Vagrant ist eine einfach konfigurierbare und portable Testumgebung[24], die für alle an Synfo-2015 beteiligten Entwickler die Möglichkeit bietet, Synfo-2015 schnell und einfach zu testen, ohne einen Server einrichten zu müssen. Vagrant ermöglicht, über Konfigurationsdateien eine virtuelle Maschine zu erstellen, die unabhängig davon, ob der Entwickler mit Linux, Mac, oder Windows arbeitet, eine einheitliche Testumgebung bereit stellt. Vagrant wurde im Laufe der Entwicklung des Projektes Synfo-2015 hinzugefügt, um eine schnell und einfach aufzusetzende Testumgebung zu ermöglichen. Vagrant greift nicht in die Projektstruktur ein, was ein späteres Entfernen einfach gestalten würde. Für die Entwickler bedeutet die Benutzung von Vagrant, dass sie sich keine Gedanken über die Einrichtung eines lokalen Servers mit allen damit zusammenhängenden Konfigurierungen machen müssen, sondern lediglich Vagrant starten müssen, um arbeiten zu können. Entwickler, die kein Interesse haben, Vagrant zu benutzen oder deren Hardware es nicht erlaubt, eine virtuelle Maschine zu starten, können trotzdem an Synfo-2015 arbeiten, ohne Vagrant zu benutzen.

[24] HashiCorp, *Vagrant - Homepage*

4 Konzeption und Implementierung

Nachdem im Vorhergehenden die Herausforderung analysiert und der Auswahlprozess der anzuwendenden Technologien dargestellt wurden, widmet sich dieses Kapitel der Konzeption und Umsetzung von Synfo-2015. Für die Erstellung des Konzeptes war es zunächst nötig, die modulare Struktur des Synfo-Systems festzulegen. Als Beispiel dafür, wie die Modularität bei Yii2 umsetzbar sein könnte, wurde bei einer ersten Recherche ein kleines CMS entdeckt, das über GitHub als Open Source verfügbar ist. Das System "yii2-start" von Vova07[25] ist ein sehr einfaches CMS, das in verschiedene Module aufgeteilt ist. Dieses CMS ist, zusätzlich zu der von Yii2 gestellten, noch unvollständigen Dokumentation von Yii2, die Quelle der Inspiration für die modulare Struktur. Die Analyse dieses Projektes diente dazu, Strukturen genauer zu erfassen und Inspirationen für Synfo-2015 zu sammeln. Vor diesem Hintergrund fiel es leichter, ein neues CMS zu konzipieren und zu entwickeln.

Um einen Überblick über Synfo-2015 zu bieten, werden in Kapitel 4.1 zunächst kurz die bereits implementierten Funktionen des Synfo-2015-CMS beschrieben. Darauf aufbauend wird in Kapitel 4.2 erklärt, wie die Modularität bei Synfo-2015 definiert ist. Anschließend wird in Kapitel 4.3 erläutert, wie die verschiedenen Gruppen und die dazugehörige Rollen- und Rechteverteilung funktionieren.

4.1 Das Synfo-2015-CMS

Zum jetzigen Zeitpunkt ist Synfo-2015 bereits ein CMS, bei dem die wichtigsten Grundfunktionen implementiert sind. Es soll allerdings mit der Zeit wachsen und weitere Funktionalität bieten können. Für die visuelle Darstellung des Backends des

[25] *vova07/yii2-start.*

CMS wurde das quelloffene, unter der MIT-Lizenz zur freien Verwendung stehende Layout für WebApps "AdminLTE" von Almsaeed Studio[26] genutzt. Dieses bietet unter anderem eine grundsätzliche Struktur und Anordnung von Inhalten. Mit der Verwendung der CSS- und Javascript-Dateien von AdminLTE wird das Layout für das Backend von Synfo umgesetzt. Bevor Synfo-2015 an einen ersten Kunden verkauft wird, wird allerdings das Layout noch besser an das Produkt angepasst werden müssen. Der Screenshot 4.4 auf Seite 37 zeigt die Startseite nach dem Einloggen in das Backend der Webseite.

Synfo-2015 enthält bereits zentrale Funktionen. So können Benutzer verwaltet und in verschiedene Gruppen eingeteilt werden. Diese Benutzer haben unterschiedliche Rechte, und es werden ihnen so in ihrem Zugangsbereich und bei möglichen Aktionen klare Grenzen gesetzt. Auf das Benutzersystem von Synfo-2015 wird in Kapitel 4.3 auf Seite 42 weiter eingegangen[27].

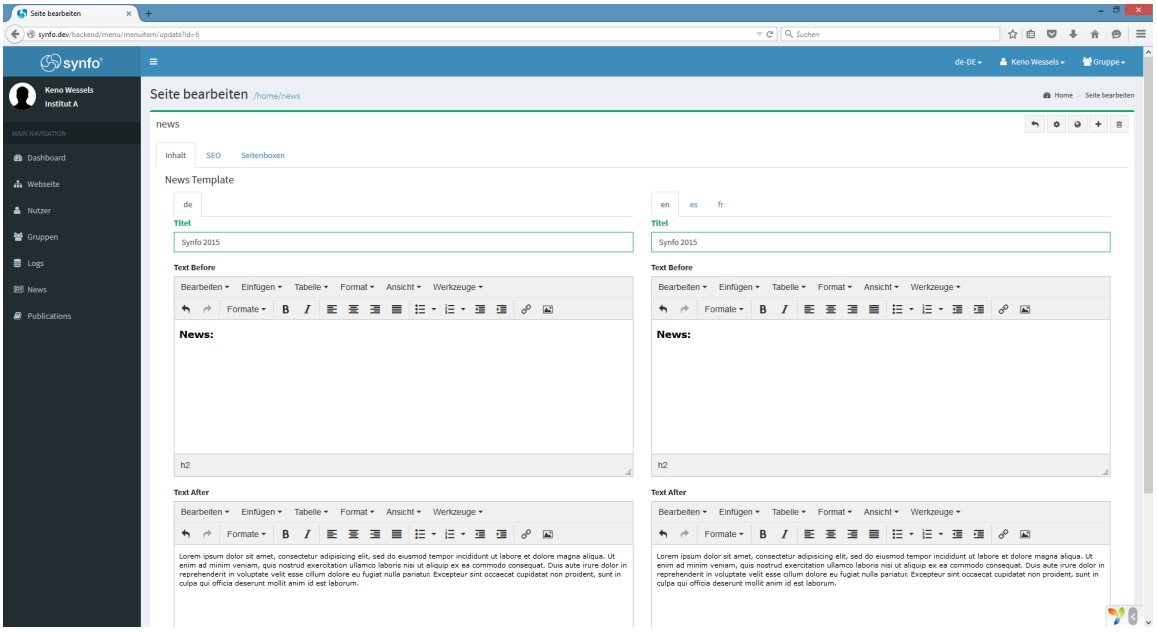

Abbildung 4.1: Synfo-2015 - Backend - Inhalt für Menüpunkt bearbeiten

Nutzer können im Backend je nach Berechtigung die Webseitenstruktur bearbeiten und Inhalte pflegen. Dafür können die Templates für einen Menüpunkt eingestellt

[26] Almsaeed Studio, *Almsaeed Studio - Homepage*

[27] Weitere Screenshots zum Verwaltungsbereich des CMS sind in den Materialien zu finden, Abbildungen: A.1, A.2, A.3, A.4, A.5

werden und jeweils genau diejenigen Inhalte verändert werden, die für das eingestellte Template vorgesehen sind. Synfo-2015 stellt bereits zwei Templates bereit: das “Main-Template“ und das “News-Template“. Diese beiden Templates dienen allerdings aktuell lediglich dem Testen der Funktionalität und bilden noch nicht den gesamten Funktionsumfang, der von diesen Templates zu erwarten wäre. Im weiteren Verlauf der Entwicklung von Synfo-2015 soll es möglich sein, ein “Standardlayout“ mit “Standardtemplates“ für jeden Kunden bereit zu stellen. Da der erste Kunde ein angepasstes Layout gebucht hat, wird deshalb für diesen direkt ein solches mit angepassten Templates erstellt. Deshalb konnten aus Zeitmangel noch keine Standardlayouts und Standardtemplates erstellt werden. Abbildung 4.1 zeigt die Bearbeitung des Inhaltes für das News-Template.

Wie in Abbildung 4.1 auch zu sehen ist, können bei Synfo-2015 Inhalte von den Benutzern in verschiedenen Sprachen bearbeitet werden. Auf dem Screenshot sind die Sprachen Deutsch, Englisch, Spanisch und Französisch zu sehen. Das gilt nicht nur für Templates, sondern auch für Inhalte aus anderen Modulen wie zum Beispiel dem News-Modul. Die Sprachen, die von den Nutzern bearbeitet werden dürfen, werden in der Konfigurationsdatei im Synfo-Skelett definiert. So ist es möglich, für jeden Kunden beliebig viele Sprachen anzubieten, ohne den Quellcode anpassen zu müssen. Diese Einstellungen lassen sich auch im laufenden Betrieb einer Synfo-2015-Website ändern (siehe Listing A.1 Synfo 2015 - params.php Zeilen 21 bis 25 auf Seite 58). Wenn eine Sprache entfernt oder hinzugefügt wird, dann wird sowohl das Backend als auch das Frontend automatisch darauf angepasst. Die Inhalte einer im Verlauf entfernten Sprache bleiben allerdings bestehen, so dass sie, falls die Sprache wieder hinzu gefügt wird, nicht zwingend mit neuen Inhalten gepflegt werden müsste. Um diese Anpassungsfähigkeit zu erreichen, muss nicht nur der Kern entsprechend flexibel programmiert sein, sondern auch alle Module, die das Pflegen von Inhalten ermöglichen, die in verschiedenen Sprachen verfügbar sein müssen. Deshalb wurde für alle Entwickler, die an Synfo-2015 arbeiten, eine Konvention eingeführt, wie mit Mehrsprachigkeit umzugehen ist. So sollen alle von Benutzern gepflegten Inhalte in einer Tabelle gespeichert werden, die ein Feld besitzt, um die Sprache zu definieren. Um die Redundanz gering zu halten, werden Inhalte, die nicht sprachabhängig sind, in einer separaten Tabelle behandelt.

id	groupid	active	date	updated_At	...
1	1	1	1436010551	1436010571	...
2	3	0	1436010565	1436010571	...
3	5	1	1436010571	1436010571	...

Tabelle 4.1: Datenbanktabelle "news_news" aus dem News-Modul

newsId	lang	title	text	updated_At	...
1	de	Neuigkeit eins	Neues Gebäude eröffnet...	1436010571	...
1	en	News one	New Building opens...	1436010571	...
3	en	News two	Professor Muster leaves...	1436010565	...

Tabelle 4.2: Datenbanktabelle "news_news_lang" aus dem News-Modul

Die beiden Tabellen 4.1 und 4.2 zeigen beispielhaft die Datenbanktabellen "news_news" und "news_news_lang" von Synfo-2015. Diese beiden Tabellen sind verknüpft und bilden so die in dem CMS eingetragenen Neuigkeiten ab. Der Primary Key in der Tabelle "news_news_lang" besteht aus den beiden Feldern "newsId" und "lang". So wird sichergestellt, dass es zu jeder News jede Sprache nur einmal gibt. Der Foreign Key der Tabelle "news_news_lang" ist das Feld "newsId" und zeigt auf das Feld "id" der Tabelle "news_news".

4.2 Definition der modularen Struktur

Modulare Software birgt viele Vorteile: Sie kann beliebig erweitert werden, es gibt meist einen Kern und einzelne Module. Der Kern liefert die gesamte Information, die notwendig ist, um die Module einzubinden und gegebenenfalls zu verbinden. Die Module kommunizieren über eine definierte Schnittstelle miteinander[28]. Diese Schnittstelle wird definiert durch die von außen zugänglichen Funktionen eines Moduls[29]. Die Module stellen je nach Struktur und Grad der Modularisierung einzelne Funktionalitäten oder eine Anhäufung von Funktionalitäten dar.

Es ist möglich, Modularität in Software in unterschiedlichen Graden umzusetzen. Je höher der Grad der Modularisierung, desto mehr einzelne Module gibt es, die

[28] Frielingsdorf u. a., 2006, S. 216.
[29] Fildebrandt, 2012, S. 18.

unabhängig voneinander weiterentwickelt werden können[30]. Ein hoher Grad der Modularisierung ist in vielen Bereichen sicherlich sinnvoll, für Synfo-2015 wird allerdings kein hoher Grad der Modularisierung angestrebt, die Module sollen einen auch für den Kunden möglichst verständlichen Zweck erfüllen. Bei zu starker Modularisierung können aber die einzelnen Funktionalitäten so sehr getrennt werden, dass für kleine Änderungen gegebenenfalls wiederum Änderungen an vielen verschiedenen Modulen vorgenommen werden müssten[31].

Die Idee der Modularisierung bei Synfo-2015 besteht darin, für verschiedene Kunden bei geringem Aufwand mit dem gleichen Kern unterschiedliche Systeme zu erstellen. Um dies zu realisieren, werden die Funktionalitäten modularisiert, die für verschiedene Kunden variabel sind. Der Kern besteht hier aus dem zuvor besprochenen, selbst konzipierten CMS für gruppenbasiertes Bearbeiten. Die einzelnen optionalen Module bilden aktuell jeweils einzelne Bereiche der Website ab, die nicht zu einem Standard-CMS gehören. Aktuell sind die Module “News“ und “Publication“ implementiert. Jedes dieser Module bietet dem Benutzer des CMS im Backend einen Bereich zum Bearbeiten und Verwalten von entsprechenden Inhalten. Weitere Module können im späteren Verlauf für Synfo konzipiert und eingesetzt werden. Die Module können dabei verschiedene Zwecke erfüllen. So könnten beispielsweise Frontend-Elemente wie Javascript-Slider[32] als Synfo-Modul erstellt und als externes Modul hinzu gefügt werden. Ein solcher Slider könnte dann bei dem Erstellen von neuen Templates helfen. Auch eine Schnittstelle, die bei Bedarf für einen einzelnen Synfo-2015-Kunden konzipiert werden könnte, würde als neues Modul implementiert werden. Eine solche Schnittstelle könnte zum Beispiel das Laden von weiteren Benutzerdaten aus einer anderen schon bestehenden Datenbank des Instituts ermöglichen.

Um die modulare Struktur bei Synfo-2015 zu erreichen, wurde das System zunächst in zwei Teile geteilt, den Synfo-Kern und die externen Synfo-Module. Der Synfo-Kern sollte unabhängig von den externen Module sein, weshalb diese ausgelagert wurden. Im Laufe des Entwicklungsprozesses wurde allerdings klar, dass diese Struktur so nicht ausreicht. Der Kern muss für jeden Kunden identisch sein. Änderungen, die an dem Kern gemacht werden, gelten für alle Kunden, wobei unterschiedliche Kunden

[30] Ludewig und Lichter, 2013, S. 411.
[31] Fildebrandt, 2012, S. 15.
[32] Ein Slider ist im Web-Bereich ein Element im Frontend, das Inhalte, meistens Bilder, entweder durch einen automatischen Intervall oder durch Aktion des Webseitenbetrachters wechselt.

verschiedene Versionen des Synfo-Kerns installiert haben können. Beispielsweise kann bei einem Kunden noch eine ältere Version von Synfo installiert sein, als bei einem anderen Kunden. Es ist wichtig, dass der Kern selbst nicht für einzelne Kunden so angepasst wird, dass er für andere Kunden nicht weiter verwendet werden kann. Um also auch den Kern unabhängig zu machen und auf die wichtigsten Funktionen zu reduzieren, wurden die Konfigurationsdateien in ein eigenes Repository ausgelagert, das Synfo-Skelett. So teilt sich das Projekt nun in folgende Bereiche:

- **Synfo-Skelett.** Das Skelett beinhaltet den Grundaufbau, der für jede Installation von Synfo notwendig ist. Hier werden Basis-Einstellungen für Synfo-2015 unternommen und definiert, welche Module geladen werden sollen.
- **Synfo-Kern.** Der Synfo-Kern beinhaltet die komplette Logik, die notwendig ist, um die Modularität zu ermöglichen. Außerdem enthält der Synfo-Kern die Webseite mit CMS für gruppenbasiertes Bearbeiten.
- **Synfo-Module.** Die einzelnen Module können zu Synfo hinzugefügt werden, ohne dass an dem Synfo-Kern etwas angepasst werden muss. Synfo-Module können das System um vielfältige Funktionen erweitern, für die erste Version von Synfo-2015 sind aber nur folgende Synfo-Module vorgesehen: synfo-news, synfo-project, synfo-event und synfo-publication.

Nachfolgend wird näher auf das Synfo-Skelett eingegangen, um die grundsätzliche Struktur von Synfo-2015 zu verdeutlichen. Anschließend wird die Schnittstelle für die einzelnen Module definiert, um eine Basis zu schaffen für den Synfo-Kern als grundlegendes CMS. Abschließend werden die optionalen, externen Synfo-Module beschrieben.

4.2.1 Das Synfo-Skelett als Strukturelement zur Ermöglichung von Modularität

Das Synfo-Sekelett besteht aus Konfigurationsdateien für Yii2, Composer, Vagrant und Synfo-2015. Die Konfiguration für Yii2 ist auf mehrere Ordner aufgeteilt, damit die verschiedenen Bereiche der Website einzeln konfiguriert werden können. Sie sind folgendermaßen untergliedert:

- In dem Ordner **Common** wird alles konfiguriert, was für das Frontend, Backend und die Konsole gleichermaßen gilt, also zum Beispiel die Daten zur Verbindung mit der Datenbank oder auch das Einstellen des Yii2 “AuthManagers“ und des “AssetManagers“.

- In dem Ordner **Frontend** wird in den Einstellungsdateien zum Beispiel der “urlManager“ von Yii2 konfiguriert und auf eine Synfo-Klasse verwiesen, die das Routing im Frontend übernimmt.

- In dem Ordner **Backend** wird unter anderem konfiguriert, welcher Theme in dem gesamten Backend-Bereich genutzt werden soll (siehe synfo-themes 4.2.3.9).

- In dem Ordner **Console** werden Einstellungen für die Ermöglichung von Konsolenbefehlen für Synfo-2015 gemacht. Die erstmalige Initialisierung von Synfo-2015 geschieht über die Konsole. Jedes Modul kann eigene Konsolenbefehle implementieren, zum Beispiel für das Hinzufügen von Rechten und Rollen zu dem RBAC System (siehe siehe Kapitel 4.3.2 “RBAC – Role-Based Access Control“). Um diese Konsolenbefehle der Module nutzen zu können müssen sie aber zu den Einstellungen in dem “Console“ Ordner hinzugefügt werden.

In diesen vier Ordnern wird durch die entsprechenden Konfigurationsdateien die grundsätzliche Konfigurierung von Yii2 unternommen. So wird dort unter anderem die Datenbankanbindung definiert, aber auch, welches Modul wo eingebunden werden soll. So kann ein Modul beispielsweise nur für das Backend oder das Frontend genutzt werden. “Routing“, “Themes“ oder der “Errorhandler“ werden in diesen Konfigurationsdateien definiert. Yii2 bietet die Möglichkeit hier auch eine eigene Konfigurationsdatei “params.php“ für das System zu implementieren. Einstellungen aus dieser Datei betreffen nicht direkt die Konfiguration von Yii2, sondern werden bei Synfo-2015 verwendet, um Einstellungen des Kerns und der einzelnen Module zu ermöglichen. So wird hier festgelegt, wie viele und welche Sprachen das System abdecken soll, während der Synfo-Kern und auch die Synfo-Module über diese Einstellung nur abfragt, welche Sprachen möglich sind.

Durch die Möglichkeit, in dem Synfo-Skelett die Einstellungen sowohl für Yii2 als auch für den Synfo-Kern und die Synfo-Module anzupassen, wird erreicht, dass diese nicht für jeden Synfo-Kunden einzeln angepasst und individualisiert werden müssen. Zusätzliche Kundendaten werden aktuell in einem gesonderten Modul umgesetzt,

das in der jeweiligen Kopie des Synfo Skelettes liegt. Diese zusätzlichen Kundendaten beinhalten unter anderem das Layout der Seite sowie eigene Templates.

4.2.2 Definition einer Schnittstelle für Synfo-Module

Um für Synfo-2015 ein CMS als Gesamtbild aus einzelnen und austauschbaren Modulen bilden zu können, musste für die Kommunikation zwischen den Synfo-Modulen eine Schnittstelle definiert werden, die diejenigen Funktionen beinhaltet, die jedes Synfo-Modul haben muss, um innerhalb des CMS angesprochen werden zu können.

In Yii2 wird die Kommunikation zwischen dem Yii2-Framework und seinen Modulen dadurch ermöglicht, dass eine Klasse `yii\base\Module`[33] bereitgestellt wird. Die Modularität wird in Yii2-Systemen dadurch ermöglicht, dass jedes Yii2-Modul eine eigene Klasse `Module` in der Root-Ebene enthält, die von `yii\base\Module` erbt. Dadurch erkennt das Yii2-Framework, dass es sich um ein Modul handelt und kann dieses entsprechend einbinden. Da also jedes Yii2-Modul diese Klasse `Module` enthält, wird diese auch als Schnittstelle zwischen dem Yii2-Framework und den Modulen eingesetzt.

Yii2 bietet damit die Möglichkeit zur Erstellung unabhängiger Module, die miteinander kommunizieren und arbeiten und ein System bilden können. Dabei werden die einzelnen für das System nötigen Module entweder in ein großes Modul zusammengefasst oder durch direkte Verweise miteinander verknüpft. Das bedeutet allerdings, dass die verknüpften Module nicht mehr vollständig unabhängig voneinander sind, sondern jetzt nur noch funktionieren, wenn alle durch Verweise verbundenen Module verfügbar sind. Der Vorteil der Unabhängigkeit der Module geht damit verloren. Als Ergebnis ist das System nicht mehr so flexibel anpassbar.

Der Synfo-Kern ist aus solchen verknüpften Modulen zusammengesetzt (siehe Kapitel 4.2.3). Das Zusammensetzen aus unabhängigen Modulen zu einem einheitlichen Synfo-Kern wäre eine nicht zu bewältigende Aufgabe gewesen, da die Implementierung einer Schnittstelle, die diese Flexibilität erlauben würde, zu viel Zeit gekostet

[33] Quellcodeausschnitte, welche in den Textfluss eingebearbeiet sind, werden in dieser Arbeit auf diese Art und Weise Formatiert: `Beispiel`

hätte. Außerdem würde die Implementierung jedes einzelnen Moduls in dem Synfo-Kern sehr viel mehr Arbeit kosten. Um allerdings die hier angestrebte Flexibilität zu erreichen, wurden nur die dafür notwendigen abhängigen Module in einem Synfo-Kern zusammengefasst. Über diesen Kern hinausgehende Module sollten unabhängig und dadurch flexibel und austauschbar bleiben können. Dieses konnte durch die Einführung einer eigenen, einheitlichen Synfo-2015-Schnittstelle erreicht werden, die für jedes Modul ausführbare Funktionen definiert. Damit die über den Kern hinausgehenden externen Module unabhängig bleiben, dürfen sie nur über diese Schnittstelle angesprochen werden.

Zu der Konzeption von Synfo-2015 bestand eine wesentliche Entscheidung darin, wie stark die Module voneinander getrennt sein sollten, also vor allem, wie viele der Synfo-Module in den Synfo-Kern gehören und wie komplex sich somit der Synfo-Kern gestalten würde. Die Überlegung, den Synfo-Kern auf ein einzelnes Modul zu reduzieren, wurde schnell verworfen, da die Module in dem Synfo-Kern auch eine Struktur für den Quellcode bieten. Die Funktionen der Synfo-Schnittstelle sind auch in den Modulen des Synfo-Kerns verfügbar. Anders als die unabhängigen externen Synfo-Module bleiben sie aber durch die direkten Verweise aufeinander voneinander abhängig. Durch die Definition des Synfo-Kerns ist dies aber nicht negativ zu bewerten.

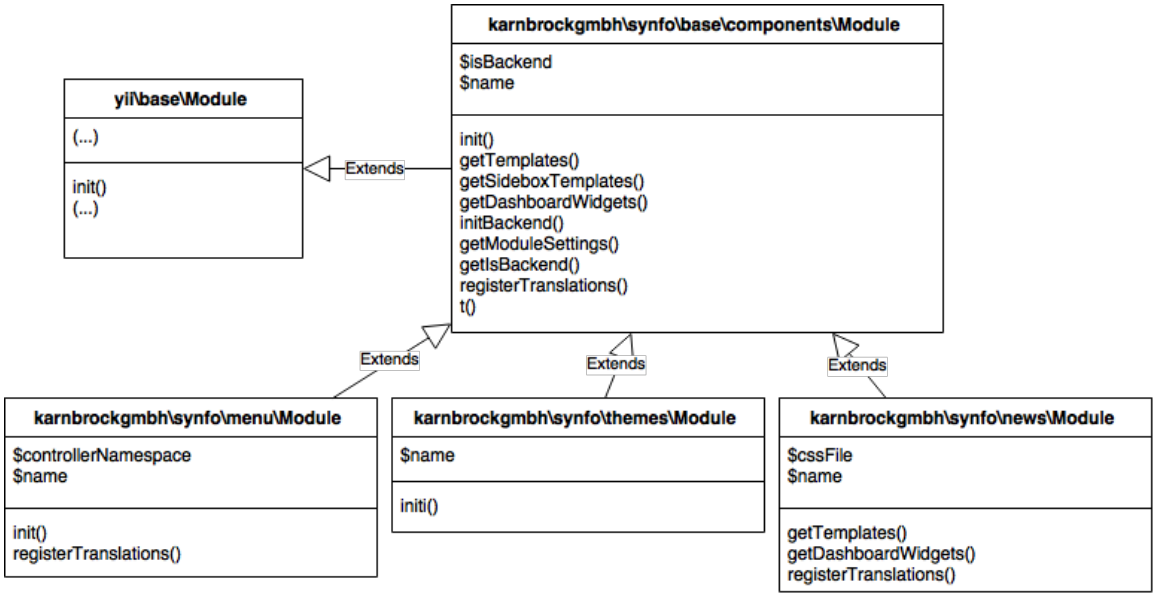

Abbildung 4.2: UML-Klassendiagramm: Struktur der Schnittstelle von Synfo-2015

Um die eigene Schnittstelle für Synfo-Module umzusetzen, wurde die Klasse `yii\base\Module` für Synfo-Module erweitert. Auf diese Weise ist die Klasse `karnbrockgmbh\synfo\base\components\Module` entstanden (siehe Abbbildung 4.2, oben rechts), die die Schnittstelle zwischen allen Synfo-Modulen definiert. Sie erbt von der Yii2-Klasse `yii\base\Module` (siehe Abbbildung 4.2, oben links) und erweitert diese um spezifische Synfo-Schnittstellen-Funktionen.

Dadurch, dass die `Module`-Klassen aller Synfo-Module sowohl aus dem Synfo-Kern, als auch aus externen Modulen nicht mehr direkt von der Yii2-Klasse `yii\base\Module` erben dürfen, sondern von der für Synfo-2015 erweiterten Schnittstellen-Klasse[34] erben, beinhaltet jedes Synfo-Modul die definierte Synfo-Schnittstelle (siehe Abbildung 4.2, unten; beispielhaft drei Synfo-Module).

Da sich die Schnittstellen-Klasse im Synfo-Base-Modul des Synfo-Kerns befindet, kann die Schnittstelle um neue Funktionen erweitert werden, ohne dass die Synfo-Module unbedingt selbst angepasst werden müssen. Anhand der statischen `getTemplates()`-Funktion wird im Folgenden die Funktionsweise der Schnittstelle beispielhaft erläutert (Siehe Listing 4.1 Zeilen 12 bis 15).

```
1  namespace karnbrockgmbh\synfo\base\components;
2  use Yii;
3  use yii\base\InvalidConfigException;
4  /**
5   * base-module.
6   */
7  class Module extends \yii\base\Module
8  {
9    /**
10    * @return array
11    */
12   public static function getTemplates()
13   {
14       return []; //Die Funktion gibt standardmäßig ein leeres
             Array zurück
15   }
16   // weitere Funktionen der definierten Synfo-Schnittstelle
17 }
```

Listing 4.1: karnbrockgmbh\synfo\base\components\Module (Ausschnitt)

[34] Die Klasse `karnbrockgmbh\synfo\base\components\Module` wird der Einfachheit halber in Zukunft Schnittstellen-Klasse genannt.

Da jedes Synfo-Modul von der Schnittstellen-Klasse erbt, kann die `getTemplates()` Funktion auch bei jedem Synfo-Modul aufgerufen werden. Wenn ein Synfo-Modul also ein Template anbietet, in diesem Beispiel das News-Modul, muss nur die Funktion `getTemplates()` in der `Module`-Klasse des News-Moduls überschrieben werden, damit ein Array mit den verfügbaren Templates zurückgegeben wird. So wird ermöglicht, dass der Synfo-Kern über die Funktion `getTemplates()` der Synfo-Module jedes Synfo-Modul anfragen kann, ob es Templates für die Website zur Verfügung stellt.

In Listing 4.2 ist ein Ausschnitt der `Module`-Klasse des synfo-news-Moduls zu sehen. Es zeigt die überschriebene statische Funktion `getTemplates()`. In diesem Fall gibt die Funktion kein leeres Array zurück, sondern definiert das Template "News", welches dieses Modul für die Webseite zur Verfügung stellt.

```
1 public static function getTemplates()
2 {
3     return [
4         'news' => 'News'
5     ];
6 }
```

Listing 4.2: karnbrockgmbh\synfo\templates\Module (Ausschnitt)

Listing 4.3 zeigt die im Synfo-Kern enthaltene Funktion `getAvailableTemplates()`, die alle Templates von allen im Synfo-Skelett definierten Modulen über die Funktion `getTemplates()` lädt.

```
1 public static function getAvailableTemplates()
2 {
3     $templates = [];
4     foreach(Yii::$app->params['templating']['templates'] as
          $ext){
5         $class = BaseHelper::getModuleClassbyName($ext);
6         $templates = array_merge( $templates, array_keys(
              $class::getTemplates()));
7     }
8     return $templates;
9 }
```

Listing 4.3: karnbrockgmbh\synfo\menu\models\PageTemplate (Ausschnitt)

In der ersten Zeile wird die Funktion `getAvailableTemplates()` als statische Funktion deklariert. Sie nimmt keine Parameter an. Anschließend wird ein leeres Array initialisiert, `$templates`, das zum Ende der Funktion zurückgegeben wird. In Zeile 4 wird eine Foreach-Schleife gestartet. Diese iteriert durch die in den Einstellungen des Synfo-Skeletts definierten Module, welche Templates anbieten dürfen. In jedem Schleifendurchlauf steht in der Variable `$ext` der Modulname des entsprechenden Moduls.

Der Synfo-Kern weiß nicht, welche Synfo-Module verfügbar sind, da diese nicht über den Synfo-Kern definiert werden. Um den Synfo-Kern also durch Module beliebig erweiterbar zu gestalten, darf der Synfo-Kern keine direkten Verweise auf die einzelnen Module enthalten, da sonst für jedes neue Modul der Kern angepasst werden müsste. Um vom Kern trotzdem auf externe Synfo-Module verweisen zu können, wurde die Hilfsfunktion, `getModuleClassbyName($name)` entwickelt und im Synfo-Kern implementiert. Dieser Hilfsfunktion wird als Parameter ein Modulname übergeben. Sofern das gesuchte Synfo-Modul verfügbar ist, gibt sie die `Module`-Klasse (Schnittstelle) des gesuchten Moduls zurück.

In Zeile 5 wird also die statische Funktion `getModuleClassbyName($name)` ausgeführt, um die entsprechende `Module`-Klasse des angefragten Moduls zu erhalten. Dieser Funktion wird der Modulname des aktuellen Schleifendurchlaufs übergeben, in diesem Beispiel das synfo-news-Modul, `getModuleClassbyName("news")` gibt wiederum die Module-Klasse (Schnittstelle) des News-Moduls zurück. Diese wird in der Variable `$class` gespeichert.

Im Anschluss werden in Zeile 6 über den Aufruf `$class::getTemplates()` die Templates aus der Schnittstelle des News Moduls geladen und zum Array `$templates` hinzugefügt. Dieses Array wird abschließend von der Funktion `getAvailableTemplates()` zurückgegeben. Diese Funktion ist eine Hilfsfunktion zum Anzeigen aller verfügbaren Templates bei dem Bearbeiten von Menüpunkten im Synfo-CMS.

Weitere Schnittstellenfunktionen, die für Synfo-2015 zur intermodularen Kommunikation entwickelt wurden, sind: `getSideboxTemplates()`, `getDashboardWidgets()` und `getModuleSettings()`. Zum Abfragen dieser Funktionen gibt es ähnliche Funktionen im Synfo-Kern wie die `getAvailableTemplates()`-Funktion.

4.2.3 Der Synfo-Kern als grundlegendes CMS

Der Synfo-Kern selbst besteht aus Modulen, wie sie von Yii2 definiert werden. Diese Module sind allerdings alle in einem einzigen Repository zusammengestellt. Dadurch wird der gesamte Synfo-Kern zusammen versioniert und die Versionsnummer wird für den gesamten Synfo-Kern hochgezählt. Das hat den Vorteil, dass der Kern als solcher immer zusammenhängend ist. Allerdings ist es in dieser Konstellation schwieriger, einzelne Synfo-Kern-Module auszutauschen. Um dem zu begegnen, ist der Kern jedoch so aufgebaut, dass die einzelnen Kern-Module ohne Probleme in einzelne Repositories ausgelagert werden und somit als vollwertige Synfo-Module fungieren könnten. Zwischen Modulen im Synfo-Kern wurden keine Abhängigkeiten definiert, da sie alle Teil des Kerns sind und somit immer gemeinsam verfügbar sind. Deshalb haben die einzelnen Module im Synfo-Kern keine eigenen Versionsnummern.

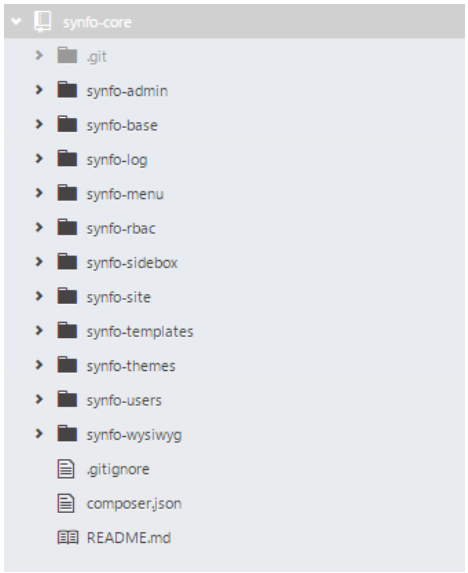

Abbildung 4.3: Die Ordner von Synfo-Kern

Abbildung 4.3 zeigt die Dateien und Ordner, die sich auf der Root-Ebene des Synfo-Kerns befinden. Der versteckte Ordner ".git" und die Datei ".gitignore" sind Dateien für die Versionsverwaltung mit Git. Die Datei "composer.json" definiert, wie das Synfo-Kern-Repository als Composer-Paket heißt und welche Abhängigkeiten es hat. Die "Readme.md"-Datei enthält Informationen zum Synfo-Kern und eine Installationsanleitung für die Entwickler. Bei den restlichen Ordnern, die in der Abbildung

4.3 zu sehen sind, handelt es sich um die Module, welche sich im Synfo-Kern befinden. Auf diese wird nachfolgend im Einzelnen eingegangen. Die Darstellung erfolgt in alphabetischer Reihenfolge und nicht nach Relevanz:

4.2.3.1 synfo-admin

Das synfo-admin-Modul wird nur im Backend eingebunden und enthält backendspezifische Daten, zum Beispiel die Fehlerseite, welche im Backend gezeigt werden soll, wenn ein Fehler auftritt oder eine Seite nicht gefunden werden kann. Außerdem implementiert es das “Dashboard“, die Startseite für das Backend. Weiterhin enthält es einen Controller, der von anderen Modulen erweitert werden kann, um als Controller im Backend eingesetzt zu werden. Dieser Controller ist deshalb sinnvoll, weil er allen Backend-Controllern bestimmte generalisierte “Actions“ zur Verfügung stellen kann. So wird für jeden Backend-Controller eine “Fileupload“-Action bereit gestellt. Diese Action kümmert sich um das Hochladen von Dateien.

4.2.3.2 synfo-base

Basisfunktionalitäten laufen über das Synfo-Base-Modul. Es enthält vor allem Klassen, die in anderen Modulen geerbt werden. Dieses Modul ist sozusagen die tiefste Ebene bei Synfo-2015. Eine wichtige Klasse aus diesem Modul wurde in dem Kapitel 4.2.2 Definition einer Schnittstelle für Synfo-Module auf Seite 30 schon beschrieben.

4.2.3.3 synfo-log

Das synfo-log-Modul, fängt Aktionen der Nutzer im Backend ab und speichert, was der Nutzer zu welchem Zeitpunkt gemacht hat. Das wird durch eine statische Log-Funktion, die von den Entwicklern überall implementiert werden kann, um Aktionen der Nutzer zu dokumentieren, ermöglicht. Vorerst wurde eine nicht komplexe Log-Funktion für das Backend implementiert, welche lediglich den Benutzer, die Gruppe, in der er sich gerade befindet, und einen von dem Entwickler gesetzten Text speichert. Diese Funktion ist für eine erste Version vollkommen ausreichend, da es aktuell noch keine Anfrage von Kunden für dieses Feature gibt. Sobald eine komplexere Version

von einem Kunden angefragt wird, kann das Modul im Synfo-Kern für alle Synfo-Kunden entsprechend angepasst werden.

4.2.3.4 synfo-menu

Das synfo-menu-Modul kümmert sich um die Menüstruktur und die Menüführung. In diesem Modul befinden sich die Models, Views und Controller zur Bearbeitung der Menüpunkte im Backend. So kann der Nutzer neue Menüpunkte erstellen oder schon vorhandene verwalten. Den Menüpunkten können Templates zugewiesen werden. Diese Templates definieren, was auf einer Seite zu sehen ist. Um die Inhalte zu verwalten, die je nach Template unterschiedlich sind, bietet jedes Template eine View und ein Model. Diese werden von dem synfo-menu-Modul geladen.

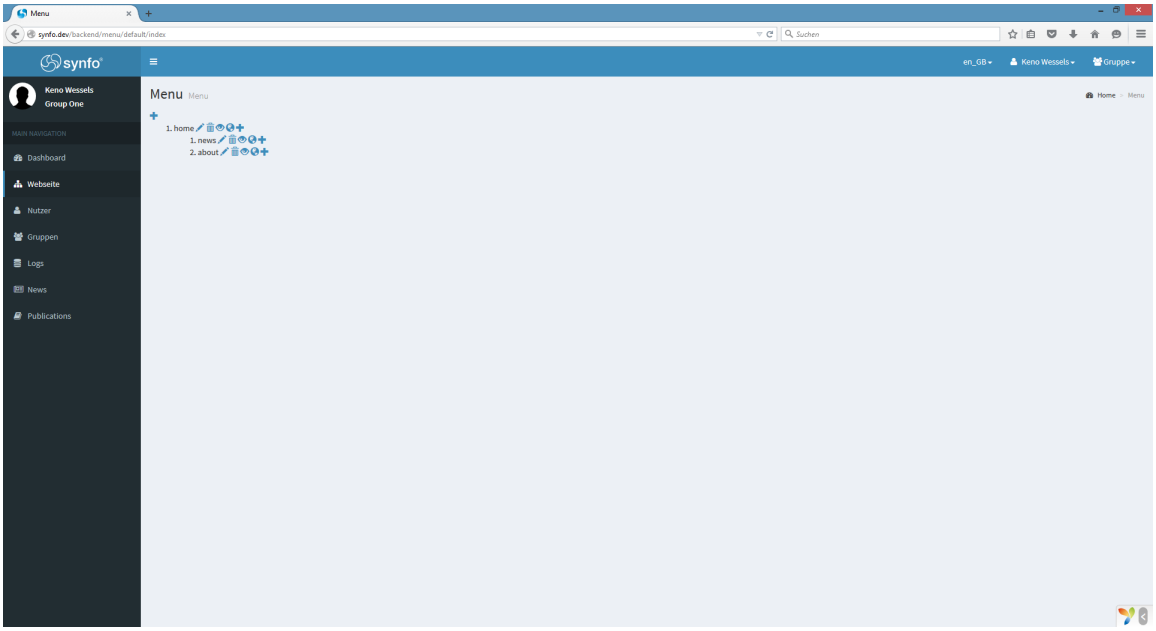

Abbildung 4.4: Synfo-2015 - Backend - Menü

In Abbildung 4.4 ist eine einfache Verwaltungsübersicht der Menüpunkte zu sehen. Bei Synfo-2015 ist es noch nicht möglich, die Reihenfolge von Menüpunkten im Nachhinein zu verändern. In Abbildung 4.1 (siehe Seite 24) ist das Bearbeiten der Inhalte eines Templates zu sehen. In dieser Abbildung ist auch zu erkennen, wie das Bearbeiten von mehrsprachigen Inhalten in Synfo-2015 abläuft. Auf der linken Seite ist

immer die in den Einstellungen definierte Standardsprache zu sehen. Auf der rechten Seite werden gleichzeitig alle weiteren Sprachen als Tabs angezeigt. Das ermöglicht dem Benutzer das direkte Übersetzen der Inhalte in andere Sprachen.

4.2.3.5 synfo-rbac

Das synfo-rbac-Modul stellt die grundlegende Struktur für die Arbeit mit RBAC bereit. Außerdem werden die wichtigsten Rechte und Rollen für die Verwendung des Backends implementiert. RBAC wird in Kapitel 4.3.2 “RBAC – Role-Based Access Control“ auf Seite 47 weiter beschrieben.

4.2.3.6 synfo-sidebox

Das synfo-sidebox-Modul implementiert die Logik für Seitenboxen. Dieses Modul ist stark abhängig von dem Modul synfo-menu. Es bietet die Verwaltung von Seitenboxen für die verschiedenen Menüpunkte. Während die Verwaltungsoberfläche für Seitenboxen von dem synfo-menu-Modul bei der Bearbeitung von Menüpunkten geladen wird, ist die Verwaltung selbst über das synfo-sidebox-Modul geregelt. Zu jedem Menüpunkt können beliebig viele Seitenboxen links und rechts hinzugefügt werden. Für Seitenboxen können eigene Templates definiert werden. So können Inhalte von Seitenboxen über Textbearbeitungsfelder, Filter und weitere Felder angepasst werden.

4.2.3.7 synfo-site

Das synfo-site-Modul wird nur im Frontend eingebunden und enthält die grundsätzliche Logik für das Frontend. Im synfo-site-Modul befindet sich unter anderem die `SiteUrlRule`, die für das Routing im Frontend zuständig ist.

Wenn ein Browser eine Seite im Frontend öffnet, so bekommt die `SiteUrlRule` von Yii2 die gesamte aufgerufene URL übergeben. Sie durchsucht dann über das synfo-menu-Modul den Seitenbaum nach dem Menüpunkt passend zu der eingegebenen URL. Wird diese gefunden, lädt die SiteUrlRule den Menüpunkt und gibt die Id des Menüpunktes an den `PageController` weiter, der sich auch in synfo-site befindet. Der

`PageController` lädt dann alle zusätzlichen Informationen zum Rendern, rendert dann mit diesen das für den Menüpunkt eingestellte Template und gibt diese an den Browser zurück.

```
1 public function parseRequest($manager, $request)
2 {
3     $menuItem = new MenuItem();
4     $params = array();
5     $path = trim($request->getPathInfo(), '/');
6     //(...)
7     $params['menuItemId'] = $menuItem->resolvePathToItem($path
          );
8     return ['/site/default/pages', $params];
9 }
```

Listing 4.4: karnbrockgmbh\synfo\site\components\SiteUrlRule (Auschnitt)

4.2.3.8 synfo-templates

Das Modul synfo-templates enthält die Logik für Templates, wie unter anderem das Laden der verfügbaren Templates, die dem Nutzer zur Verfügung stehen. Die Templates selbst kommen aus den in der Konfiguration (Synfo-Skelett) definierten Modulen. Das synfo-templates-Modul stellt auch Basistemplates bereit, um das Arbeiten mit Synfo-2015 zu ermöglichen, ohne kundenspezifische Templates zu erstellen. Um die Basistemplates zu nutzen, muss das synfo-templates-Modul in der Konfiguration als Quelle für Templates mit angegeben sein.

4.2.3.9 synfo-themes

Das Modul synfo-themes definiert die Themes für das Frontend und Backend. Zu Themes gehört das grundsätzliche Aussehen, wie Header-Bereich oder Footer-Bereich auf einer Website. Außerdem werden globale “Stylesheets“ in den Themes definiert, wie zum Beispiel die Styles von “AdminLTE“ für das Backend. Welches Theme wo genutzt wird, wird über die Einstellungen im Synfo-Skelett definiert. So kann für einen neuen Synfo-2015 Kunden ein neues Theme außerhalb des Kerns erstellt werden, das auf seine Bedürfnisse angepasst ist. Dieses Theme kann in den Einstellungen dann anstatt des Standardthemes als Theme für das Frontend eingestellt werden.

4.2.3.10 synfo-users

Das Modul synfo-users bietet im Backend einen Bereich für die Benutzerverwaltung. Dort können Benutzer hinzugefügt, entfernt und bearbeitet werden. Benutzer können außerdem inaktiv geschaltet werden. Die Authentifizierung der Benutzer wird über dieses Modul geregelt. Wenn sich ein Benutzer in das Backend einloggen möchte, so werden seine Log-In-Daten mit denen aus dem Benutzer Modul abgeglichen. Sind die Daten korrekt, darf er das Backend entsprechend der seinem Nutzerprofil zugeordneten Rechte verwalten.

Für Synfo-2015 wurde noch keine Möglichkeit eingearbeitet, dass Mitarbeiter eines Instituts ihre persönlichen Seiten über das Backend bearbeiten können. Diese Funktionalität soll im späteren Verlauf noch für das synfo-users-Modul umgesetzt werden.

4.2.3.11 synfo-wysiwyg

Das Modul synfo-wysiwyg implementiert einen "What You See Is What You Get"-Editor (WYSIYG-Editor) für Synfo. Dieser ist ein Text-Editor, bei dem der Inhalt, welchen man bei dem Bearbeiten sieht, dem entspricht, was in der Ausgabe zu sehen ist[35]. Aktuell wird hier TinyMCE als Editor für Synfo-2015 verwendet, da er bereits für das alte Synfo-2012 System genutzt wurde und gut funktioniert. TinyMCE ist ein plattform-unabhängiger, web-basierter WYSIWYG-Editor auf der Basis von Javascript[36]. Dieser Editor wird von verschiedenen Modulen für verschiedene Zwecke eingebunden (siehe Abbildung 4.1, Seite 24).

4.2.4 Externe Synfo-Module als optionale Erweiterungen – das News Modul als Beispiel

Synfo-Module befinden sich nicht in dem Synfo-Kern. Sie sind abhängig vom Synfo-Kern, aber nicht umgekehrt. Sie können definieren, mit welcher Version des Kerns sie funktionieren. Sie können außerdem mit anderen Modulen kommunizieren. Es ist zwar noch keine Verbindung zwischen verschiedenen optionalen Synfo-Modulen

[35]Oxford University Press, *Oxford Dictionaries - Homepage - definition WYSIWYG*
[36]Ephox, *Tinymce - Homepage*

implementiert, jedoch sollte eine solche Verbindung auch über die in Kapitel 4.2.2 definierte Schnittstelle umsetzbar sein.

Im Prinzip unterscheiden sich externe Synfo-Module nicht unbedingt von den Modulen im Synfo-Kern. Sie sind zwar abhängig vom Kern, aber, zumindest bis jetzt, von keinem anderen Synfo-Modul. Die externen Synfo-Module müssen, um verschiedene Funktionen und Inhalte beitragen zu können, bestimmte Regeln und Strukturen einhalten. Sie können nur über die Schnittstelle angesprochen werden.

Bislang wurden die beiden externen Synfo-Module synfo-publication und synfo-news entwickelt. Für eine erste Verkaufsversion von Synfo-2015 sind noch die beiden externen Synfo-Module synfo-event und synfo-project geplant.

Mit dem synfo-news-Modul wurde die Schnittstelle zwischen Synfo-Kern und externen Synfo-Modulen getestet. Es implementiert darüber hinaus aber noch nicht alle notwendigen Funktionen.

Das externe Synfo-Modul synfo-publication ist nicht von dem Autor dieser Arbeit implementiert worden und ist deshalb nicht Teil dieser Arbeit.

Das synfo-news-Modul implementiert eine Verwaltungsoberfläche für News in dem Backend von Synfo und stellt ein News-Template bereit. Die Verwaltungsoberfläche bietet die Möglichkeit des Erstellens, Bearbeitens und Löschens von News. Außerdem können News auch übergeordneten Gruppen vorgeschlagen werden. Damit können dann auch Kommentare über den Inhalt oder die Vollständigkeit des Inhaltes zwischen den verschiedenen Nutzern hin und her geschickt werden. Das synfo-news-Modul implementiert außerdem ein Basistemplate, mit dem es möglich ist, sich alle News im Frontend anzeigen zu lassen. Außerdem stellt es Gruppenberechtigungseinstellungen bereit, mit denen definiert werden kann, welche Gruppe welche Rechte innerhalb des synfo-news-Moduls hat.

Abbildung 4.5 zeigt die Verwaltungsoberfläche des News-Moduls. In dem oberen Teil, “Proposed News“ sind Neuigkeiten zu sehen, welche nicht von der aktuell aktiven Gruppe erstellt, aber dieser vorgeschlagen wurden. In dem unteren Teil sind alle News der aktiven Gruppe aufgelistet, um sie zu verwalten.

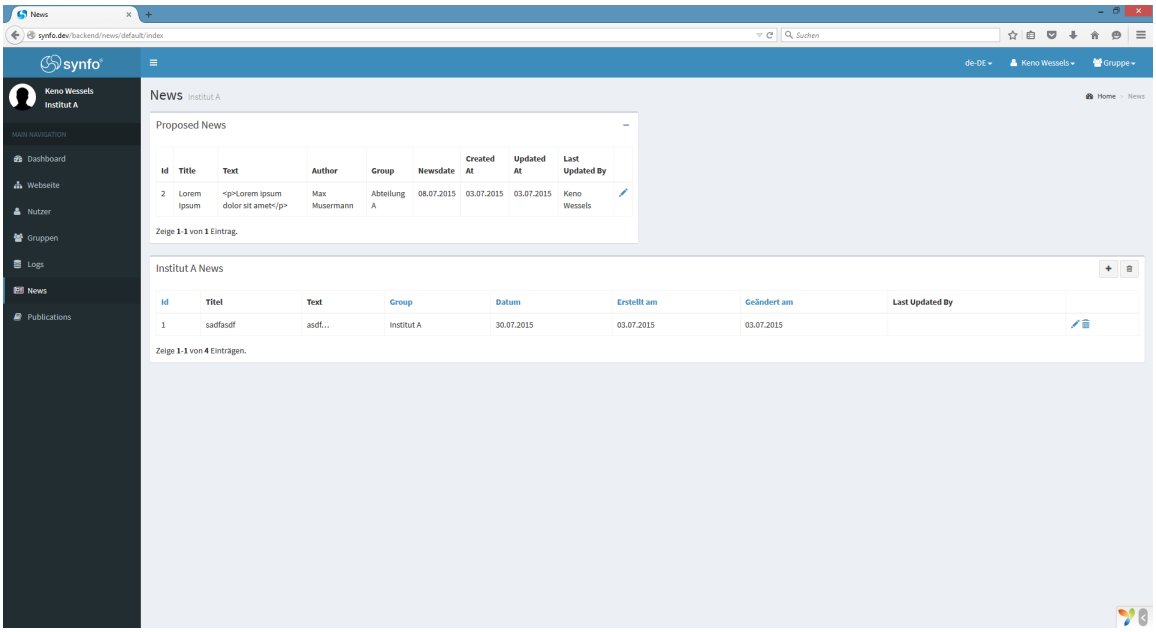

Abbildung 4.5: Synfo-2015 - Backend - News Verwaltung

Das News-Modul stellt dem Synfo-2015-CMS ein News-Template bereit. Dieses Template kann in dem CMS Menüpunkten zugeteilt werden. Es beinhaltet zwei Textfelder und ein Titelfeld, welche jeweils in allen verfügbaren Sprachen bearbeitet werden können. Außerdem zeigt es im Frontend alle verfügbaren News. Da auch das News-Template nur Testzwecken diente, gibt es noch nicht die Möglichkeit, sich nur Newseinträge bestimmter Gruppen anzeigen zu lassen.

4.3 Gruppenbasiertes Bearbeiten

Da für gewöhnlich fast alle Mitarbeiter Zugang mit unterschiedlichen Rechten zu dem System bekommen sollen, muss definiert werden, welcher Nutzer zu welchen Inhalten Zugang hat und welche Inhalte er bearbeiten darf. In dem bestehenden Synfo-2012-System wird diese Aufgabe gelöst, indem es verschiedene Zugänge für die verschiedenen Bereiche gibt. Synfo-2015 soll aber flexibler sein.

4.3.1 Gruppenstruktur

Die Institute, deren Websites die Firma Karnbrock GmbH bisher umgesetzt hat, haben verschiedene Abteilungen, mit unterschiedlichen Mitarbeitern. So ergibt sich eine Aufteilung wie sie in Abbildung 4.6 zu sehen ist. Synfo-2012 bietet für die drei in Abbildung 4.6 zu sehenden grauen Ebenen jeweils einen eigenen Zugangsbereich. Zusätzlich gibt es einen weiteren Zugangsbereich für die Verwaltung der mitarbeitereigenen Daten.

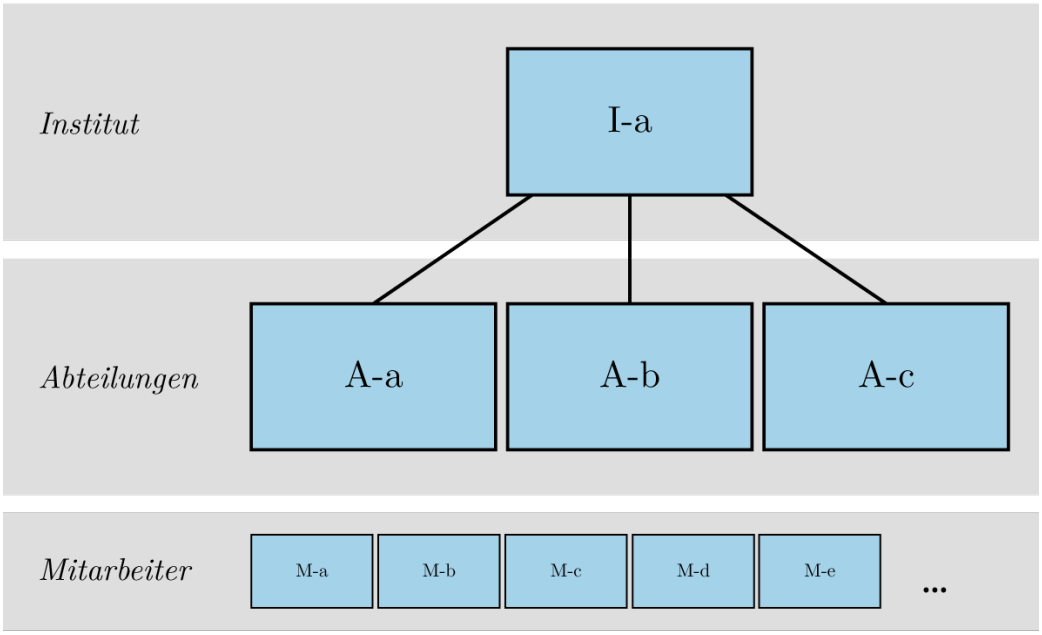

Abbildung 4.6: Aufteilung von Gruppen bei Synfo-2012

Um die hierarchische Ebenenstruktur zu verändern bzw. zu erweitern, musste bei Synfo-2012 das gesamte CMS umgeschrieben werden. Einer der bisherigen Synfo-2012-Kunden benötigte beispielsweise keine Abteilungsebene, so dass für diesen Kunden bei der Umsetzung von Synfo-2012 der Quellcode angepasst werden musste.

Bei der Neukonzeption von Synfo-2015 war zunächst geplant, das System genauso statisch aufzubauen wie bei Synfo-2012. Allerdings sollten die drei verschiedenen Zugänge zu dem Verwaltungsbereich durch nur einen Zugang ersetzt werden, um die Benutzerfreundlichkeit zu verbessern. Dies ermöglicht es dem Benutzer mit nur einem Login, zwischen verschiedenen Ebenen zu wechseln. Dann wurde aber die Idee

entwickelt, die Hierarchie der Ebenen so flexibel zu gestalten, dass, wenn man nicht mit drei Ebenen arbeiten möchte, der Quellcode trotzdem nicht neu angepasst werden müsste. Dafür hätte man die Ebene der Abteilungen so konfigurierbar machen müssen, dass man sie bei der Installation eines Synfo-2015-Systems für jeden Kunden beliebig aktivieren oder deaktivieren könnte. Damit wäre die Flexibilität gewonnen, auch ein kleineres Institut abzubilden.

Um Synfo-2015 aber auch bei einem größeren, mit mehreren Ebenen arbeitenden Institut zum Einsatz bringen zu können, wurde überlegt, wie eine oder mehrere zusätzliche Ebenen ermöglicht werden könnten. Dafür müssten sowohl die Ebenen als auch die Gruppen komplett flexibel zu gestalten sein. Sollte das gelingen, würde das neue System in Zukunft beliebig viele Ebenen mit beliebig vielen Gruppen ermöglichen und deshalb auch viele verschiedene Kunden abbilden können.

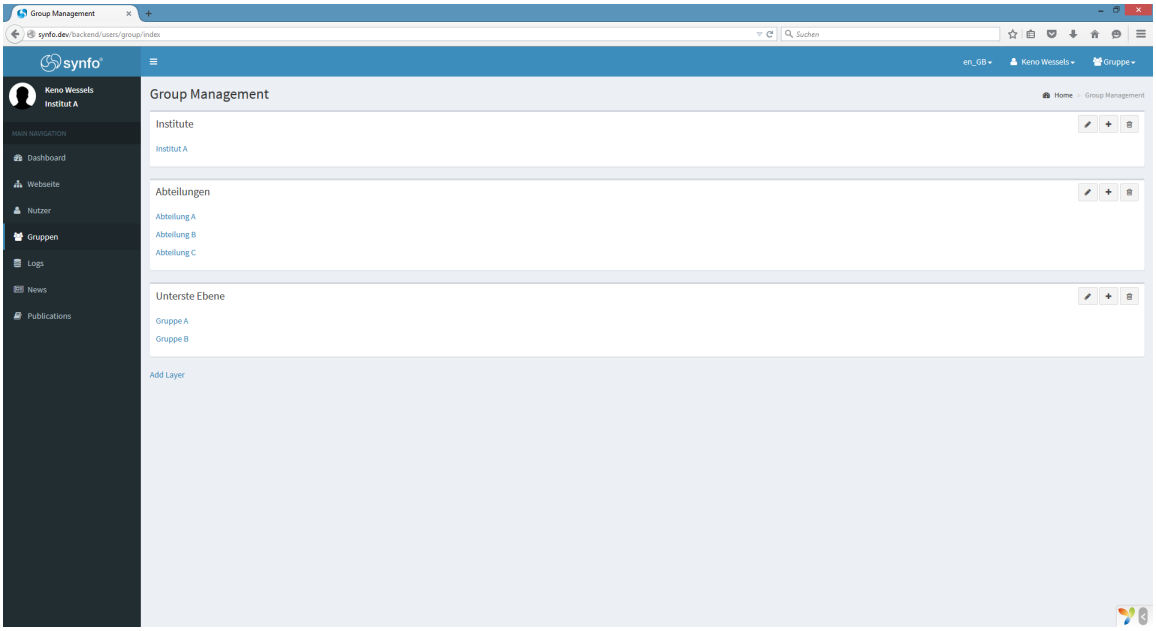

Abbildung 4.7: Synfo-2015 - Backend - Gruppenverwaltung

Für die Umsetzung einer neuen Gruppenverwaltung in dieser Größenordnung waren unterschiedliche Schritte nötig. Im Verwaltungsbereich des Synfo-2015-CMS wurde dafür zunächst ein eigener Bereich eingeführt. Es mussten Gruppen erstellt werden, die sich nach der gewünschten Funktionalität bearbeiten ließen. Dafür wurden Models jeweils für Gruppen und Ebenen erstellt und zu ihrer Bearbeitung ein Controller

eingeführt. Letzterer verfügt über eine View, die eine Bearbeitungsoberfläche für den Verwaltungsbereich des CMS darstellt (siehe Abbildung 4.7). Mit Hilfe des Controllers und der View ist es möglich, die Gruppen und Ebenen zu verwalten. Der Controler ermöglicht es auch, die hierarchischen Strukturen dieser Gruppen zu definieren, also festzulegen, welche Gruppen als übergeordnet gelten sollen. Somit ist das Gruppensystem so flexibel, dass jederzeit neue Ebenen und Gruppen hinzugefügt werden können.

Wie in Abbildung 4.8 zu sehen, ist es außerdem möglich, dass Gruppen gleichzeitig zwei übergeordnete Gruppen haben können. Gruppe "**Ag-e**" ist Untergruppe von "**A-b**" und "**A-c**". Dieses Feature ist für Synfo-2012 noch nicht angefragt worden, es ist allerdings durchaus denkbar, dass es beispielsweise Projektgruppen gibt, die Teil zweier Abteilungen sind.

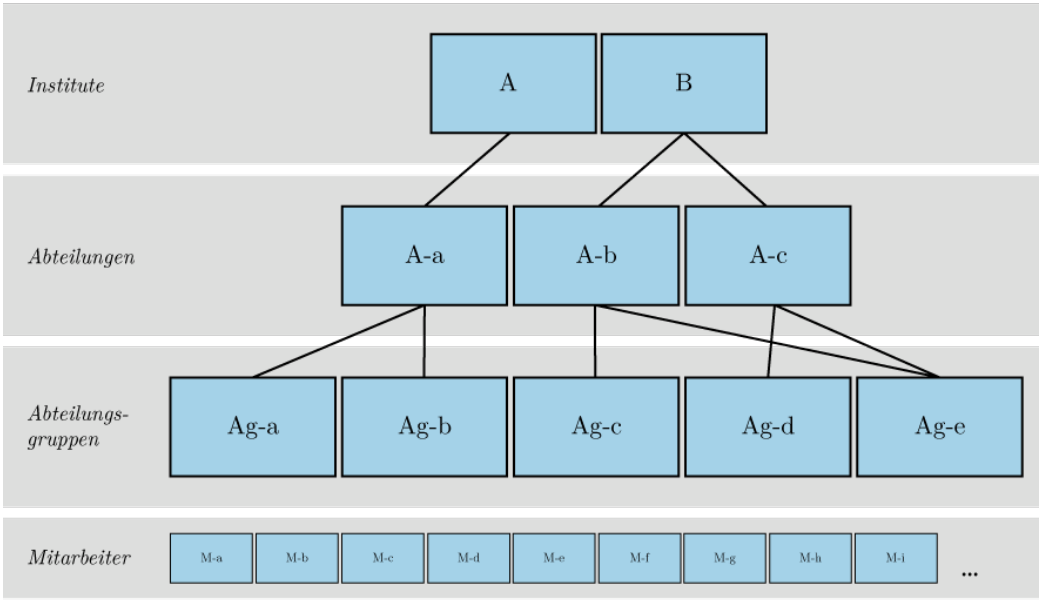

Abbildung 4.8: Beispielhafte Aufteilung von Gruppen bei Synfo-2015

Im nächsten Schritt müssen die Benutzer mit diesen Gruppen verknüpft werden können. Damit geht einher, dass sie für jede dieser Gruppen eine eigene Rolle bekommen. Das ist notwendig, da sie gleichzeitig verschiedenen Gruppen zugerechnet werden und dabei verschiedene Rechte haben können. Anschließend wurde für das Backend implementiert, dass der Benutzer die Gruppe und damit gegebenenfalls auch seine Rolle im Verwaltungsbereich wechseln kann. Der Benutzer kann also über ein Auswahlfeld

im Backend wählen, in welcher Gruppe er aktiv sein möchte, um die Inhalte der Internetseite zu ändern. Abbildung 4.9 zeigt dieses Auswahlmenü im Backend.

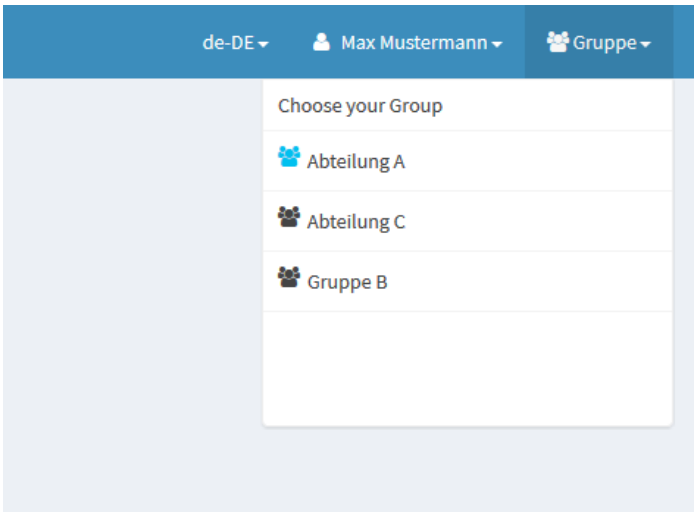

Abbildung 4.9: Auswahl der "aktiven Gruppe" bei Synfo-2015

Max Mustermann, der Benutzer aus diesem Beispiel, ist Mitglied in drei Gruppen (siehe Abbildung A.4, Seite 57). In der "Abteilung A" und "Abteilung C" ist er "Superadmin", wohingegen er in der Gruppe "Gruppe B" nur die Rolle "User" einnimmt. Im Backend kann er auswählen, welche Gruppe für ihn aktuell aktiv sein soll. In Abbildung 4.9 ist er beispielsweise in der Gruppe "Abteilung A" eingeloggt. Da er in dieser Gruppe Superadmin ist, darf er alle Inhalte dieser Gruppe bearbeiten.

Es muss gesichert sein, dass alle die Gruppen betreffenden Inhalte mit den Gruppen verknüpft sind, damit je nach Gruppe, in der sich der Nutzer einloggt, auch nur die entsprechenden Inhalte verfügbar und verwaltbar sind. Inhalte können jetzt also nur noch gruppenweit verwaltet werden. Außerdem muss definiert werden, welche Gruppe Zugriff auf welchen Teil des Menübaums der Webseite hat bzw. welchen Teil der Internetseite sie verändern darf. Des weiteren werden Inhalte – wie beispielsweise News aus dem News-Modul – mit den Gruppen verbunden. Es werden also News, wenn sie von Benutzern erstellt werden, mit der Gruppe verknüpft, in der sich der Benutzer aktuell befindet. Benutzer aus anderen Gruppen haben somit keine Möglichkeit, diese News zu bearbeiten.

Im folgenden Schritt werden die Rechte für die Gruppen festgelegt, um bestimmte Aktionen durchführen zu können. So kann zum Beispiel ganzen Gruppen – und somit

allen ihren Nutzern – erlaubt oder nicht erlaubt werden, News zu erstellen und zu verwalten. Im letzten Schritt wird die Möglichkeit implementiert, dass Inhalte wie zum Beispiel News, an übergeordnete Gruppen vorgeschlagen werden können.

Jeder der Gruppen aus Abbildung 4.8 können also unterschiedliche Rechte zugewiesen werden. Die Gruppe "**Ag-e**" könnte zum Beispiel die Berechtigung haben, in einem bestimmten Bereich die Inhalte im Frontend zu bearbeiten. Gleichzeitig dürfte "**Ag-e**" beispielsweise News verwalten, aber keine Publikationen.

Abschließend kann man sagen, das Synfo-2015 durch seine flexible Gruppenstruktur sehr viel skalierbarer für unterschiedliche Kunden ist.

4.3.2 RBAC – Role-Based Access Control

Synfo-2015 arbeitet nach dem Prinzip RBAC (Role-Based Access Control). Ferraiolo, Kuhn und Chandramouli[37] definieren in ihrem Buch "Role-Based Access Control", folgendermaßen:

> "Role-based access control (RBAC) is a technology that is [...] reducing the complexity and cost of security administration in large networked applications [...] by using roles, hierarchies, and constraints to organize privileges."

Das heißt, RBAC ist eine Technologie, die die Komplexität und die Kosten der Administration von großen Netzwerkanwendungen durch die Benutzung von Rollen, Hierarchien und Bedingungen zum Organisieren von Rechten reduziert. Das gilt nicht nur für Netzwerkanwendungen, sondern auch für andere Anwendungen mit vielen Nutzern, die unterschiedliche Rechte haben.

Bei Synfo kann so jeder Aktion ein eigenes Recht zugeordnet werden, das mit dieser Aktion verknüpft ist. Lediglich Nutzer, die auch dieses Recht haben, dürfen dann diese Aktion ausführen. Diese Rechte haben bei Synfo einen sprechenden Namen wie zum Beispiel: "CanAccessBackend"(darf Backend betreten), "CanUpdateMenuItem"(darf Menüeintrag bearbeiten) oder "CanViewNews" (darf News einsehen).

[37] Ferraiolo, Kuhn und Chandramouli, 2003, S. xv.

Zusätzlich zu den Rechten gibt es Rollen, mit denen Rechte gruppiert werden. Für eine Rolle lassen sich also beliebig viele Rechte definieren. Benutzer des Systems haben eine bestimmte Rolle und somit auch die Rechte, die dieser Rolle zugeschrieben sind.

Bei Synfo-2015 können Nutzer allerdings auch in unterschiedlichen Gruppen sein und in den verschiedenen Gruppen jeweils eine andere Rollen einnehmen. Um dies zu verdeutlichen, greifen wir hier zu einem Beispiel:

Die Benutzerin Erika Musterfrau ist in leitender Position in der Abteilung Musterabteilung tätig. Also bekommt sie in der Gruppe Musterabteilung die Rolle Administrator (Admin) und darf alle Inhalte dieser Gruppe verwalten und löschen. Sie ist allerdings zusätzlich die Assistentin der Leitung des Instituts Musterinstitut. Also erhält sie in der Gruppe Musterinstitut die Rolle Redakteur, damit sie auch hier Inhalte bearbeiten darf. Je nachdem, welche Gruppe Erika Musterfrau gerade ausgewählt hat, kann sie nun unterschiedliche Inhalte bearbeiten. Erika Musterfrau kann zum Beispiel in der Musterabteilung Beiträge der Gruppe Musterabteilung löschen. In dem Musterinstitut ist ihr allerdings nicht erlaubt Beiträge des Musterinstituts zu löschen, hier darf sie nur bearbeiten.

4.4 Workflow

Synfo-2015 ist als System so aufgebaut, dass im weiteren Verlauf viele Entwickler parallel daran arbeiten können. Um das möglich zu machen und damit der Quellcode einheitlich bleibt, gibt es zum einen Konventionen, die von allen Entwicklern eingehalten werden müssen. Zum anderen gibt es verschiedene Repositories, die auf GitHub gehostet werden. Die Entwickler sind auf GitHub in verschiedene Gruppen aufgeteilt. Diese Gruppen definieren die Zugriffsrechte auf die Repositories. So gibt es beispielsweise die Gruppe “Synfo-Core-Dev“. Die Benutzer, die dieser Gruppe zugeordnet sind, dürfen die Repositories “synfo-core“ (Synfo-Kern) und “synfo“ (Synfo-Skelett) bearbeiten. Die Gruppe “Synfo-Core-Read“ erlaubt ihren Mitgliedern dagegen nur die beiden Repositories “synfo-core“ und “synfo“ zu lesen. Zuletzt gibt es noch die Gruppe “Synfo-Module-Dev“, die alle Synfo-Module bearbeiten darf. Durch diese Aufteilung der Zugriffsrechte bleibt die Kontrolle über das System bei den Entwicklern, die den Kern bearbeiten dürfen. Entwickler, die an den Modulen arbeiten, müssen sich an die

verfügbaren Schnittstellen richten und sind auf diese Weise gezwungen, sich an die Konventionen zu halten.

Wenn für bestimmte Funktionalitäten doch etwas am Kern geändert werden müsste, so hat der Modulentwickler zwei Möglichkeiten: Er kann entweder den Kernentwicklern einen Vorschlag machen, was geändert werden müsste, damit diese Funktion möglich wäre. Die Kernentwickler müssten diese Funktion dann, falls sie sie für sinnvoll erachten, implementieren. Die andere Möglichkeit besteht darin, dass der Modulentwickler den Kern "forkt", also eine Kopie des Repositories erstellt, in der er auch Schreibrechte hat. In dieser Kopie des Repositories könnte der Modulentwickler dann die ihm fehlende Funktionalität implementieren. Wenn diese abgeschlossen ist, würde der Modulentwickler ein "pull-request" stellen, also die Anfrage an das entsprechende Repository richten, seine Änderungen zu übernehmen. Über diesen pull-request können die Kernentwickler zusammen mit den Modulentwicklern auf der Plattform GitHub diskutieren. Falls noch Modifikationen fehlen sollten, die vor der Übernahme des Änderungsvorschlages in das Original-Repository gemacht werden müssten, könnten diese noch zu dem Pull-request hinzu gefügt werden. Wenn die Kontrolle abgeschlossen ist, wird der Pull-request von den Kernentwicklern angenommen und somit in das entsprechende Repository eingefügt.

Durch diese klare Trennung zwischen Synfo-Kern und externen Synfo-Modulen bleibt die Kontrolle über das System bei einer überschaubaren Zahl von ausgewählten Personen, was einen einheitlicheren Quellcode für den Synfo-Kern sicherstellen sollte. Trotzdem können sich Entwickler, die keinen direkten Zugriff auf den Synfo-Kern haben, durch die Erstellung von "pull-requests" kontrolliert an der Entwicklung des Synfo-Kerns beteiligen. So gibt es mehr Kommunikation zwischen den Entwicklern, wodurch im Idealfall auch der Quellcode optimiert werden kann.

5 Fazit

Im Rahmen des vorliegenden Projekts sollte während einer halbjährigen Entwicklungszeit ein CMS für Institute an Universitäten entstehen. Dieses System sollte für die Verwaltung der Inhalte auf die Strukturen universitärer Institute angepasst sein. Außerdem sollte das System auf aktueller Technologie aufbauen.

Am vorläufigen Ende dieses Prozesses ist mit Synfo-2015 ein CMS entwickelt worden, das zum jetzigen Zeitpunkt noch nicht für den gleichen Zweck wie seine Vorgängerversion, Synfo-2012, eingesetzt werden kann, weil ihm im aktuellen Entwicklungsstadium noch einige Funktionen fehlen. Das neu konzipierte CMS bietet aber einige grundsätzliche Vorteile gegenüber Synfo-2012: Zum einen handelt es sich bei Synfo-2015 um ein System, das mit dem Blick auf die erforderlichen Grundfähigkeiten als Einheit entwickelt wurde. Dabei wurde die Implementierung komplett unabhängig von Synfo-2012 auf einer technologisch neuen Basis vollzogen. Um das neu entwickelte Synfo-2015 in diesem Prozess konsistent zu halten, wurden dabei keine Quellcodeausschnitte von Synfo-2012 übernommen. Synfo-2012 hatte sich dagegen nicht systematisch sondern ad hoc entwickelt, wie es die Anforderungen der Kunden im konkreten Fall jeweils bedingten. Dies hatte zur Folge, dass zum einen der Quellcode von Synfo-2012 nicht mehr übersichtlich war. Zum anderen mussten aber auch die gleichen Funktionen für unterschiedliche Synfo-2012 Kunden mehrfach implementiert werden.

Synfo-2015 hingegen basiert auf aktuellen Technologien und ist von Grund auf einheitlich konzipiert und entwickelt worden. Es ist auf die gezielte und systematische Weiterentwicklung des Systems auch in Zukunft und durch andere bzw. mehrere Entwickler ausgelegt. Als Programmiersprache wird in beiden Versionen PHP eingesetzt – eine der gängigsten Programmiersprachen, die eine möglichst breite Anwendbarkeit und Verständlichkeit des Systems gewährleistet. Auch in der Firma Karnbrock GmbH wird routinemäßig mit PHP gearbeitet, sodass mit Synfo-2015 auch auf die bereits

bestehenden Kompetenzen der Entwickler der Firma Karnbrock GmbH aufgebaut werden kann und kostspielige und zeitintensive Weiterbildungen erspart bleiben.

Im Gegensatz zu Synfo-2012, bei dem kein Framework genutzt wurde und der Quellcode nicht strukturiert ist, wurde für Synfo-2015 auf der Basis des Frameworks Yii2 eine modulare Struktur eingeführt, die es ermöglicht, externe Module für Synfo-2015 zu entwickeln. Externe Synfo-Module haben den Vorteil, dass der Synfo-Kern unabhängig von ihnen weiterentwickelt werden kann und auf das reduziert bleibt, was ihn als Kern ausmacht. So sollte Synfo-2015 auch für neue Entwickler schnell verständlich sein.

Zugleich konnte damit auch eine Struktur für den Synfo-Kern definiert werden, die es ermöglicht, das CMS zu strukturieren. Schon jetzt bietet das CMS, das im Synfo-Kern umgesetzt wurde, die wichtigsten Funktionen. In dem Backend des CMS können Menüpunkte definiert werden, denen Templates zugeteilt werden können. Diese Templates können von den Nutzern mit Inhalten gefüllt werden. Es gibt außerdem bereits ein externes Modul, welches das Verwalten von News erlaubt.

Allerdings mussten im Rahmen der hier vorgestellten Entwicklungsarbeit Prioritäten gesetzt werden. Das bedeutet, dass einige Funktionen nicht abschließend ausgearbeitet werden konnten und andere noch neu zu entwickeln sein werden. So wird die Verwaltung von Menüpunkten noch einmal überarbeitet werden müssen, da sie noch nicht alle Funktionen abbildet, die zu einer lückenlosen Verwaltung gehören. Man kann beispielsweise noch keine Menüpunkte verschieben. Gleichwohl sind alle diese Funktionen im Ansatz mitgedacht und können von der bereitgestellten Struktur ausgehend problemlos implementiert werden.

Ebenfalls hervorzuheben ist das für Synfo-2015 neu entwickelte komplex angelegte Rechtesystem, das die Strukturen der für universitäre Institute typischen Hierarchien in angemessenerer komplexer Form abbilden kann. Während bei dem Vorgängermodell noch drei Zugangsbereiche benötigt werden, je einer für die drei Ebenen, können nun differenzierte Zugänge für eine theoretisch nicht begrenzte Zahl von Gruppen in dem CMS erstellt und verwaltet werden. Der Log-in ist aber nicht mehr direkt an die Gruppe gebunden. Diese Gruppen können hierarchisch angeordnet werden und jeweils anderen Gruppen untergliedert sein. Sie können gesonderte Berechtigungen für bestimmte Bereiche des CMS zugeteilt bekommen. Innerhalb dieser Gruppen können

einzelnen Nutzern noch einmal unterschiedliche Rechte zugewiesen werden. Diese Struktur ermöglicht es, über nur einen Zugangsbereich zum Backend verschiedene, individuell angepasste Zugriffsrechte bereitzustellen.

Synfo-2015 ist also im Vergleich zu Synfo-2012 sehr viel skalierbarer angelegt, sodass es perspektivisch auch möglich ist, je nach Bedürfnis des Kunden deutlich mehr als die in dem Synfo-2012-CMS möglichen drei Ebenen einzuführen.

Für die weitere Entwicklung am System Synfo-2015 wird auch auf die Zusammenarbeit mit den Mitarbeitern der Karnbrock GmbH gesetzt. Dafür ist es notwendig, dass die Systemarchitektur von allen beteiligten Entwicklern nachvollzogen und akzeptiert werden kann. Sie müssen sich bei der Arbeit mit Synfo-2015 im Rahmen der vorgegebenen Regeln bewegen, mit den eingeführten Technologien arbeiten und sich an definierte Konventionen halten, insbesondere in den Punkten, in denen diese sich von Synfo-2012 unterscheiden. Erste Erfahrungen konnten in dieser Hinsicht bereits gesammelt werden, da die Kollegen der Firma Karnbrock GmbH parallel zur Entstehung dieser Bachelorarbeit begonnen haben mit Synfo-2015 zu arbeiten. Ein Kollege hat bereits zwei externe Module entwickelt (“synfo-event“ und “synfo-project“), welche sehr ähnlich zu dem “synfo-news“-Modul funktionieren. Ein weiterer Entwickler arbeitet derzeit am Synfo-Kern und hier an der Menüpunktverwaltung. Positiv zu bewerten ist, dass sie gut mit dem System zurechtgekommen sind. Nachdem anfangs noch viele Verständnisfragen beantwortet und Unklarheiten beseitigt werden mussten, bearbeiten beide Entwickler ihre Aufgabenbereiche mittlerweile selbständig. Auf der Basis des von mir entwickelten Synfo-2015 werde ich in der Zukunft mit drei Kollegen an der Fertigstellung einer ersten stabilen Kunden-Version arbeiten. Ziel ist es, das erste Synfo-2015 im November 2015 in Betrieb zu nehmen.

Auch wenn es etwas dauern wird, bis Synfo-2015 das alte System Synfo-2012 in vollem Funktionsumfang ersetzen wird, zeigt sich, dass es für die Firma Karnbrock GmbH – auch aus der Sicht des Autors – richtig war, in das System Synfo-2015 zu investieren. Die Einrichtung eines CMS für einen neuen Kunden wird sehr viel systematischer und daher auch effektiver und mit weniger Arbeitsaufwand ablaufen. Das Updaten der verschiedenen verkauften und betreuten Synfo-2015-Systeme wird problemlos und schnell erfolgen können. Selbst der Einsatz des Synfo-2015-CMS für andere Kundenstämme der Firma Karnbrock GmbH ist dann denkbar. Mit leichten Anpassungen am

Synfo-Kern sollten sich auch Kunden mit einem wesentlich weniger komplexen Benutzersystem abbilden lassen können. Damit bietet Synfo-2015 eine wichtige Grundlage für die geschäftliche Zukunft der Firma Karnbrock GmbH.

6 Anhänge

A Material

A.1 Synfo-2015 CMS Screenshots

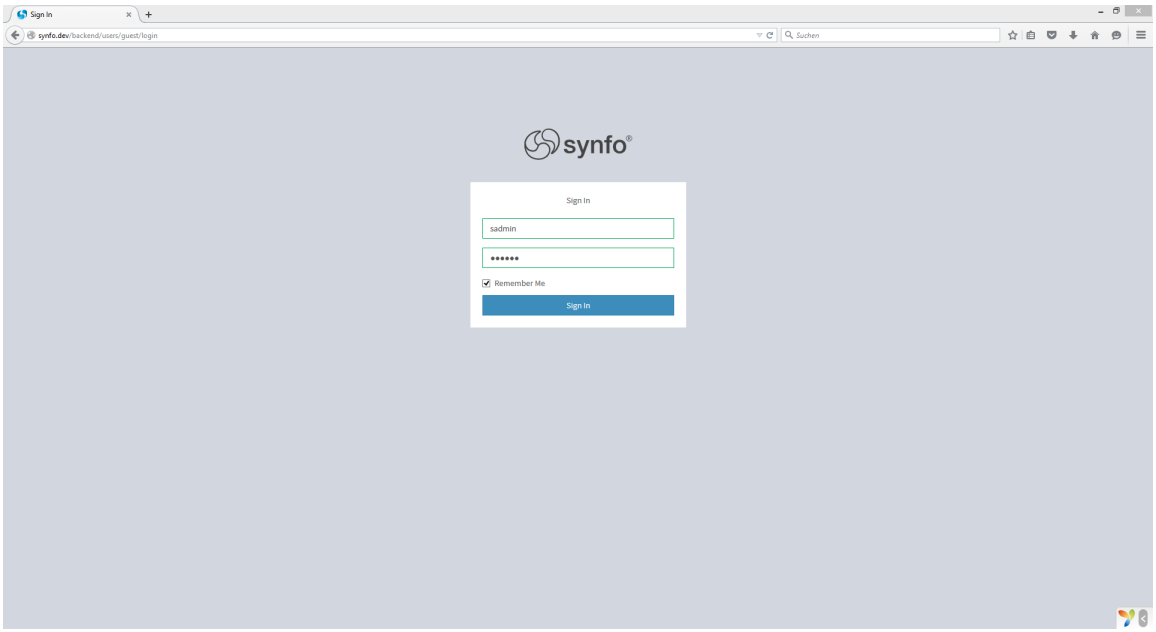

Abbildung A.1: Synfo-2015 - Backend - Log In

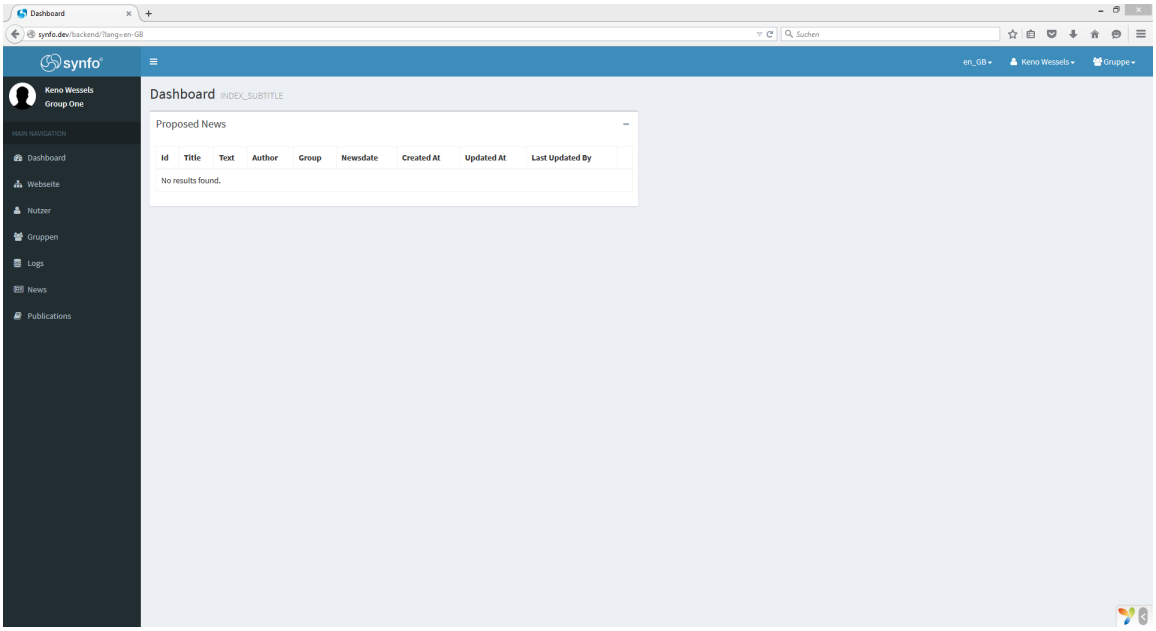

Abbildung A.2: Synfo-2015 - Backend - Dashboard

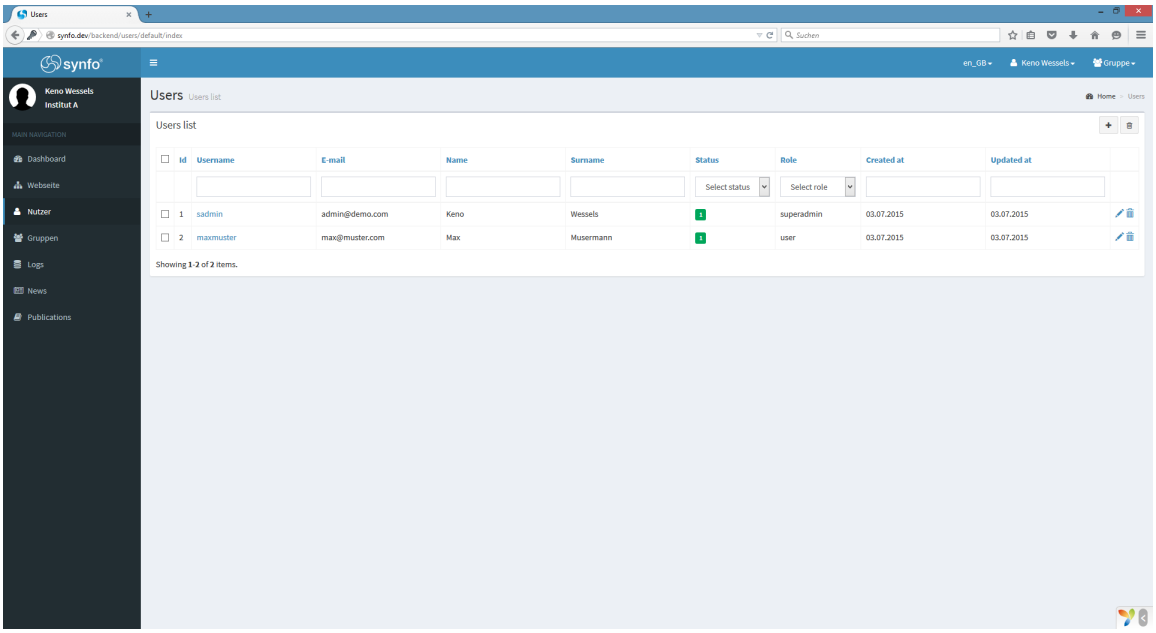

Abbildung A.3: Synfo-2015 - Backend - Benutzerverwaltung

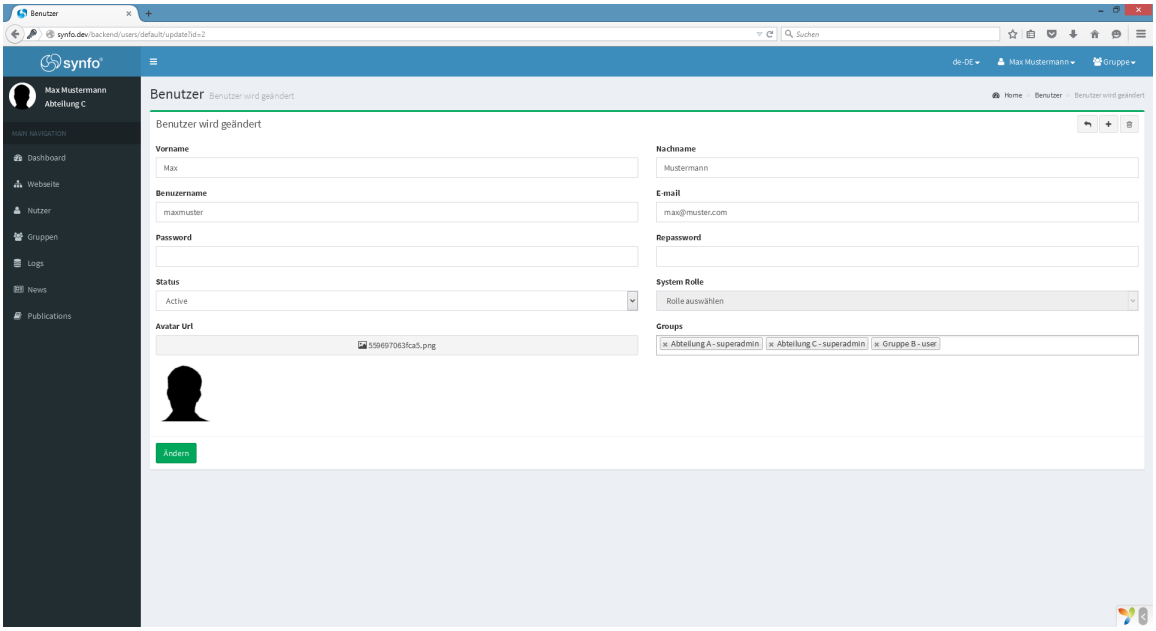

Abbildung A.4: Synfo-2015 - Backend - Benutzerbearbeitung

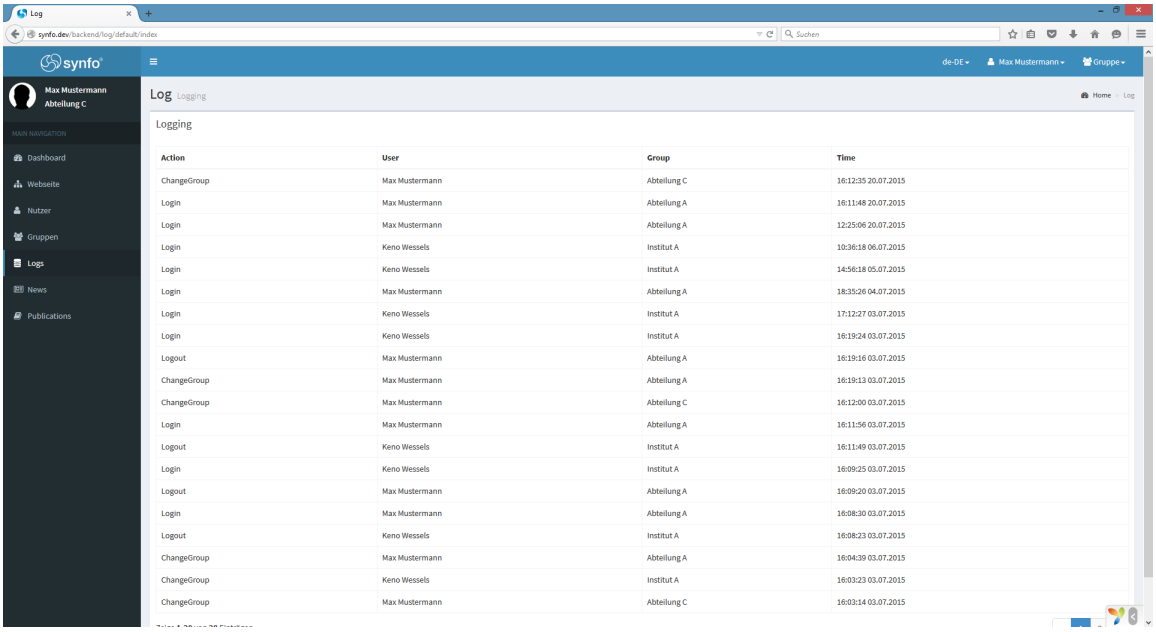

Abbildung A.5: Synfo-2015 - Backend - Log

A.2 Synfo-2015 Quellcode Ausschnitte

```
<?php return [
    'activeModules' => [
        'news' =>[
            'backend' => [
                'visibleInSidebar' => true,
                'sidebarIcon' => 'fa-newspaper-o',
                'sidebarUrl' => '/news/default/index',
                'sidebarLabel' => 'News',
                'sidebarAccesRights' => [
                    'BAccessNews'
                ]
            ],
        ],
    ],
    'defaultLanguage' => 'de-DE',
    'languages'=> [
        'de-DE' => 'Deutsch',
        'en-GB' => 'English',
        'es-ES' => 'Espanol',
    ],
    'languagesContent'=> [
        'de-DE' => 'Deutsch',
        'en-GB' => 'English',
        'es-ES' => 'Espanol',
    ],
    'templating' =>[
        'sideboxes' =>[
            'sidebox',
            'news'
        ],
        'templates' =>[
            'news',
            'templates'
        ]
    ],
    'dashboardWidgets' =>[
        'news',
    ],
];
```

Listing A.1: Synfo 2015 - params.php

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Literatur

Adermann, Nils und Jordi Boggiano. *Composer - Introduction - Dependency management*. URL: https://getcomposer.org/doc/00-intro.md#dependency-management (besucht am 30. 06. 2015).

Almsaeed Studio. *Almsaeed Studio - Homepage*. URL: https://almsaeedstudio.com/ (besucht am 03. 07. 2015).

Bruns, Kai und Klaus Meyer-Wegener, Hrsg. (2005). *Taschenbuch der Medieninformatik*. München: Fachbuchverl. Leipzig im Carl Hanser-Verl.

Chacon, Scott und Ben Straub (2014). *Pro git*. Apress. URL: https://progit2.s3.amazonaws.com/en/2015-06-28-a1792/progit-en.562.pdf.

Eclipse Foundation. *eclipse survey 2009*. URL: http://www.eclipse.org/org/press-release/Eclipse_Survey_2009_final.pdf (besucht am 17. 06. 2015).

– *eclipse survey 2010*. URL: http://www.eclipse.org/org/community_survey/Eclipse_Survey_2010_Report.pdf (besucht am 17. 06. 2015).

– *eclipse survey 2011*. URL: http://www.eclipse.org/org/community_survey/Eclipse_Survey_2011_Report.pdf (besucht am 17. 06. 2015).

– *eclipse survey 2012*. URL: http://eclipse.org/org/community_survey/Survey_Final_Results_2012.xls (besucht am 17. 06. 2015).

– *eclipse survey 2013*. URL: http://eclipse.org/org/community_survey/SurveySummary_2013_public.xlsx (besucht am 17. 06. 2015).

– *eclipse survey 2014*. URL: http://www.eclipse.org/org/community_survey/Eclipse_Survey_2010_Report.pdf (besucht am 17. 06. 2015).

Ephox. *Tinymce - Homepage*. URL: http://www.tinymce.com/ (besucht am 04. 07. 2015).

Ferraiolo, David F., D. Richard Kuhn und Ramaswamy Chandramouli (2003). *Role-Based Access Control*. Norwood, MA, USA: Artech House, Inc.

Fildebrandt, Ulf (2012). *Software modular bauen: Architektur von langlebigen Softwaresystemen ; Grundlagen und Anwendung mit OSGi und Java*. 1. Aufl. Heidelberg: dpunkt-Verl.

Fournova Software GmbH. *Tower - Homepage - 12 Git Hosting Services Compared*. URL: `http://www.git-tower.com/blog/git-hosting-services-compared/` (besucht am 06. 07. 2015).

Frielingsdorf, Herbert u. a. (2006). *Einfache IT-Systeme*. 4. Aufl. Basiswissen IT-Berufe ... Troisdorf: Bildungsverl. EINS.

GULP Information Services GmbH. *Gulp - Homepage*. URL: `https://www.gulp.de/` (besucht am 01. 07. 2015).

Geirhos, Matthias (2015). *Entwurfsmuster: das umfassende Handbuch*. 1. Aufl. Bonn: Rheinwerk-Verl.

Git. *Git - Homepage - About*. URL: `https://git-scm.com/about` (besucht am 06. 07. 2015).

GitHub, Inc. *About Github*. URL: `https://github.com/about` (besucht am 30. 06. 2015).

HashiCorp. *Vagrant - Homepage*. URL: `https://www.vagrantup.com/` (besucht am 30. 06. 2015).

Karnbrock GmbH. *Karnbrock GmbH Referenzen*. URL: `http://karnbrock.biz/referenzen` (besucht am 13. 06. 2015).

Loeliger, J. (2010). *Versionskontrolle mit Git*. O'Reilly. URL: `https://books.google.de/books?id=cedjsbyRnJEC`.

Ludewig, Jochen und Horst Lichter (2013). *Software Engineering: Grundlagen, Menschen, Prozesse, Techniken*. 3., korrigierte Aufl. Heidelberg: dpunkt.Verl.

Maurice, Florence (2015). *PHP 5.6 und MySQL 5.7: Ihr praktischer Einstieg in die Programmierung dynamischer Websites*. 4., aktualisierte und erw. Aufl. Heidelberg: dpunkt-Verl.

Meyen, Sebastian und Rainer Stropek (2013). *Software Development Trends: Wegweisende Beiträge für eine neue IT*. Ed. 2014. ... Frankfurt am Main: Entwickler.press.

Oxford University Press. *Oxford Dictionaries - Homepage - definition WYSIWYG*. URL: `http://www.oxforddictionaries.com/definition/english/WYSIWYG` (besucht am 04. 07. 2015).

Q-Success DI Gelbmann GmbH. *w3techs - Usage of server-side programming languages for websites*. URL: `http://w3techs.com/technologies/overview/programming_language/all` (besucht am 17. 06. 2015).

Teixeira, Pedro (2012). *Professional Node. js: Building Javascript based scalable software*. John Wiley & Sons.

The PHP Group. *PHP Homepage - History of PHP*. URL: `http://be2.php.net/manual/en/history.php.php` (besucht am 30. 06. 2015).

Yii Software LLC. *Yii 2.0.0 is released*. URL: `http://www.yiiframework.com/news/81/yii-2-0-0-is-released/` (besucht am 30. 06. 2015).

– *Yiiframework - Guide - How does Yii Compare with Other Frameworks?* URL: `http://www.yiiframework.com/doc-2.0/guide-intro-yii.html` (besucht am 01. 07. 2015).

vova07/yii2-start. GitHub, Inc. URL: `https://github.com/vova07/yii2-start` (besucht am 16. 06. 2015).

Ich versichere, die vorliegende Arbeit selbstständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangaben eindeutig kenntlich gemacht.

Ort, Datum Keno Weßels