

Proseminar
Website-Management-Systeme

PHPNuke

Andreas Emrich

Technische Universität Kaiserslautern
Fachbereich Informatik
Arbeitsgruppe Softwaretechnik
Betreuer: Dipl.-Inf. Christian Stenzel, Dipl.-Inf. Markus Reitz

Inhaltsverzeichnis

1. Einführung	3
1.1 Allgemeine Einführung	3
1.2 Die Skriptsprache PHP	3
1.3 Architektur von PHPNuke	3
2. Ausgewählte Komponenten von PHPNuke	5
2.1 User-Module	5
2.1.1 Artikel	5
2.1.2 Kommentare	5
2.1.3 Umfrage	6
2.2 Verwaltungsmodule	6
2.2.1 Blöckeverwaltung	6
2.2.2 Newsletter	6
2.2.3 Allgemeine Seiteneinstellungen	6
2.2.4 Benutzerverwaltung	6
3. Erweiterungsmöglichkeiten	6
3.1 Mögliche Anwendungsgebiete für eigene Module	6
3.2 Erstellung eigener Themes	7
4. Abschließende Bewertung	7
4.1 Bewertung nach den Kriterien des Vortrags Content-Management	7
4.2 Persönliches Fazit	7
5. Literaturangaben	8

1. Einführung

1.1 Allgemeine Einführung

PHPNuke ist ein *Portal- und Redaktionssystem*, das unter der General Public License (Abk.: GPL) veröffentlicht worden ist. Portal- und Redaktionssystem heißt, dass sowohl die Nutzung der hinterlegten Inhalte durch User als auch das Generieren von Inhalten durch Administratoren direkt über das Webinterface des PHPNuke-Portals erfolgt. Die General Public License von GNU ermöglicht, PHPNuke für nicht-kommerzielle Zwecke kostenlos einzusetzen, d.h. es handelt sich hierbei um ein *Open-Source*-Produkt. Des Weiteren kann man PHPNuke auch als ein *Web-Content-Management-System* bezeichnen, wie später noch deutlicher herausgestellt werden soll. Die wesentlichen Voraussetzungen für den Betrieb eines PHPNuke-Portals sind der *Apache Webserver*, *PHP* und eine *SQL-Datenbank*.

1.2 Die Skriptsprache PHP

Die hauptsächlichliche Technologie, auf der das PHPNuke-System aufbaut ist die Skriptsprache *PHP* (Abk. für **PHP Hypertext Preprocessor**). Sie ist die mittlerweile am weitesten im Web verbreitete Skriptsprache, kann sowohl als CGI als auch als Modul in einem Webserver ablaufen. PHP4, die Version, auf der PHPNuke aufbaut, kennt noch keine Typisierung. Dies ist aber für die kommende Version 5 geplant, ebenso wie Objektorientierung mit Subtyping und Exceptions ähnlich zu Java. PHP ist auch gerade deshalb sehr beliebt, da es eine C++/Java-ähnliche Syntax aufweist.

In PHPNuke stellen sich Datenanforderung des Users durch den Einsatz von PHP folgendermaßen dar: Anstatt des direkten Aufrufs einer statischen HTML-Seite wird die Anfrage durch das entsprechende PHP-Skript im PHPNuke-System bearbeitet. In diesem Bearbeitungsprozess werden ggf. auch noch andere PHP-Skripte involviert, die ebenfalls Aktionen ausführen. Als Resultat liefert PHP eine HTML-Seite an den Browser zurück.

1.3 Architektur von PHPNuke

Die Entwickler von PHPNuke unterteilen das PHPNuke-System grob in drei Kategorien: *Module*, *Themes* und die *Datenbank*.

Ein **Modul** ist eine Einheit eines oder mehrerer *PHP-Skripte*, das *eine gewisse Funktionalität* zur Verfügung stellt. Genauer abgegrenzt ist der Begriff des Moduls von offizieller Seite nicht. Allerdings kann man folgende grundlegende Modularten unterscheiden: *User-Module* (Diese können beispielsweise einen Artikel oder einen Kommentar darstellen – im Falle des Kommentars auch noch Möglichkeit zur Interaktion bieten), *Administrationsmodule* (zur Generierung und/oder Verwaltung von Inhalten), *Layout-Module* (stellen einen Bereich im Layout des Portals dar, z.B. `header.php`, `footer.php` usw.) und *Hilfsmodule* (`counter.php` zum Zählen der Seitenaufrufe, oder `sql_layer.php` zur Anbindung an SQL-Datenbanken). Allerdings ist auch diese Einteilung schon problematisch, da User-Module wie z.B. Artikel, Skripte enthalten, die die Administrationsfunktion bereitstellen, d.h. konkret, dass bei vielen User-Modulen das Administrationsmodul bereits integriert ist. Zu beachten ist insbesondere, dass man ein Modul im System nicht direkt lokalisieren kann. Mitunter können Skripte ein und desselben Moduls über mehrere Ordner verteilt sein. Darüber hinaus muss man noch den hier besprochenen allgemeinen Modulbegriff in PHPNuke vom speziellen abgrenzen: Oft sind mit Beschreibungen in Dokumentationen nur die User-Module gemeint, die in der Navigation des Systems im Block Module angezeigt werden.

Ein **Theme** ist ein Subsystem des PHPNuke-System, welches das *Erscheinungsbild des Portals* definiert. Zum einen kann der Administrator ein Default-Theme bestimmen, das immer angezeigt wird, oder die (registrierten) Benutzer können dem Portal ein eigenes Erscheinungsbild geben, indem sie ein anderes Theme auswählen. Allerdings muss das gewünschte Theme in dem PHPNuke-System vorhanden sein. Dies bestimmt sich dadurch, ob das Theme im Themes/-Ordner des PHPNuke-Systems abgelegt worden ist.

PHPNuke benutzt eine **SQL-Datenbank** zur Speicherung der Inhalte. Als Datenbank wäre hierbei MySQL, PostgreSQL, mSQL, PostgreSQL_local, ODBC, ODBC_adabas, Interbase, Sybase u.ä. denkbar. Sie wird in PHPNuke über das Modul zur Datenbankbindung – realisiert durch das Skript `sql_layer.php` – angesprochen.

Wir wollen nun genauer auf die Interaktion im Browser, die in Punkt 1.2 bereits rudimentär beschrieben worden ist, eingehen. Bei einer Anfrage wird ein entsprechendes PHP-Skript zur Ausführung gebracht. Zunächst werden dadurch die Skripte im betreffenden Modul angesprochen, die wiederum über das Modul zur Datenbankbindung die Inhalte aus der Datenbank anfordern, um gemeinsam mit den Daten zum Layout, die aus dem aktuellen Theme gewonnen werden, eine HTML-Darstellung im Browser bewirken. Wenn nun ein Administrator Inhalte generieren will, so wählt er ein Administrationsmodul an, indem er in einem Formular des Webinterfaces des Systems Daten eingibt, die dann in die SQL-Datenbank des PHP-Nuke-Systems geschrieben werden. Die folgende Grafik veranschaulicht, wie Informationen erstellt oder angefordert, und dann anschließend dargestellt werden:

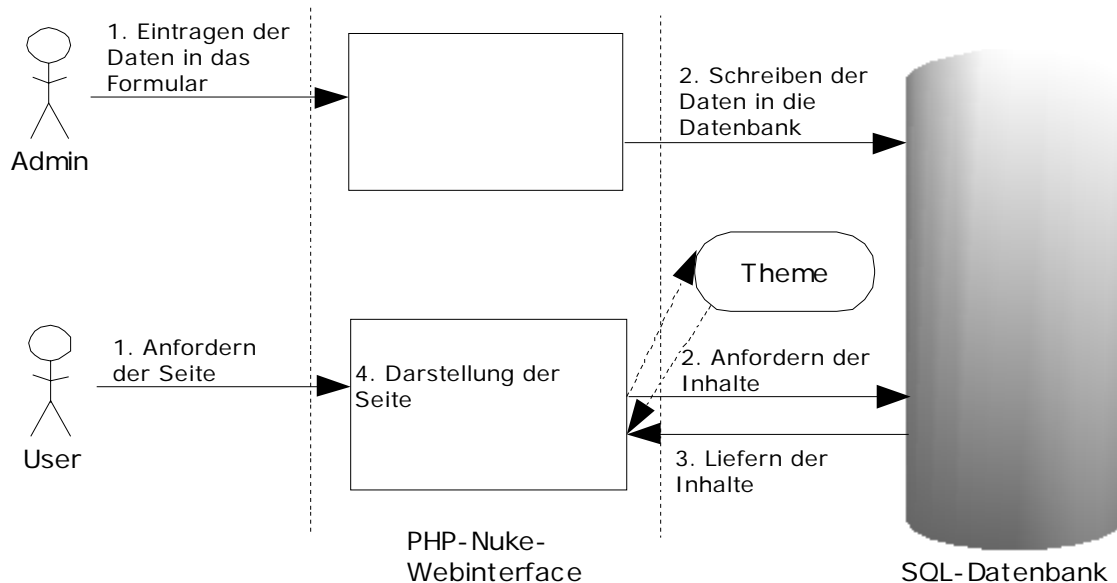


Abb. 1: Ablauf eines Skriptaufrufs in PHPNuke

Das Layout einer Seite in PHPNuke ist natürlich wie gesagt stark von den Themes abhängig. Allerdings müssen bestimmte Konventionen aus technischen oder rechtlichen Gründen eingehalten werden. So muss z.B. sinnvollerweise die Navigation angezeigt werden, um überhaupt eine benutzerfreundliche Bedienung des Portals zu gewährleisten. Des Weiteren muss z.B. im Footer des Layouts gekennzeichnet werden, dass dieses Portal mit PHPNuke erstellt worden ist. Grob vereinfacht ergibt sich somit folgendes Layout für eine Seite in einem PHPNuke-Portal:

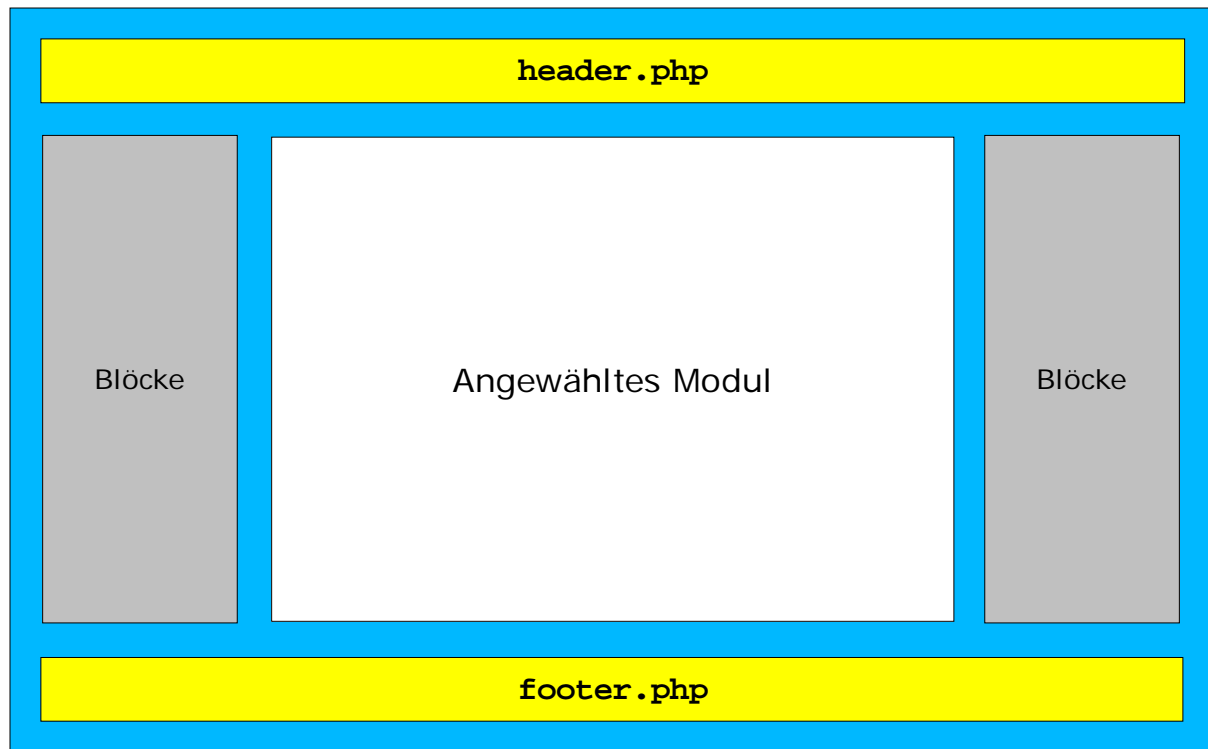


Abb. 2: Layout einer Seite des PHPNuke-Portals

Hierbei ist anzumerken, dass nur User- und Administrationsmodule wirklich angezeigt werden können. Blöcke sind in der Terminologie von PHPNuke die Navigationsbereiche, sie werden über das Administrationsmodul der Blöckeverwaltung konfiguriert, auf das in Punkt 2.2.1 eingegangen wird.

2. Ausgewählte Komponenten von PHPNuke

2.1 User-Module

2.1.1 Artikel

Ein Artikel ist das geläufigste und meist verwendete Modul im Zusammenhang mit PHPNuke oder ähnlichen Systemen. Er besteht aus einer Überschrift, einem Anreißtext und einem erweiterten Text. Zudem ist jeder Artikel in PHPNuke einem bestimmten Thema zugeordnet, das wiederum über das Administrationsmodul der Themenverwaltung definiert werden kann. User können im Webinterface aus einer Übersicht von Artikeln – dargestellt durch die Überschrift und den Anreißtext – einen Artikel auswählen, um ihn sich komplett anzuschauen. Darüber hinaus kann man über die Suchfunktion des PHPNuke-Systems einen Artikel suchen. Bei Artikeln hat der User zwei Interaktionsmöglichkeiten: Das Schreiben von Kommentaren zu dem Artikel und das Teilnehmen an einer Umfrage zu dem Artikel, falls von dem Administrator eine Umfrage für den betreffenden Artikel bereitgestellt wird.

2.1.2 Kommentare

Ein Kommentar bietet dem registrierten Benutzer die Möglichkeit zu einem Artikel seine Meinung kundzutun. Dabei gibt er in ein Formular mit den Feldern Betreff und Mitteilung seine Nachricht ein. Je nachdem wie die Benutzerverwaltung die Rechte für die registrierten

Benutzer setzt, hat der Benutzer das Recht Kommentare zu lesen, zu verfassen oder auch Kommentare anderer Benutzer zu beantworten.

2.1.3 Umfrage

Eine Umfrage besteht aus einer Frage und mehreren Auswahlmöglichkeiten. Sie ist immer mit einem Artikel verknüpft und wird bei der Anzeige des Artikels in einem Navigationsblock für Umfragen angezeigt. Der User kann hier wählen, indem er eine Auswahlmöglichkeit anklickt, oder sich das bisherige Ergebnis der Umfrage anzeigen lassen.

2.2 Verwaltungsmodule

2.2.1 Blöckeverwaltung

Die Navigationsleisten des PHPNuke-Systems befinden sich am linken und rechten Rand und sind in sog. Blöcke unterteilt. Es gibt Blöcke für die Module (gemeint sind die User-Module), den Administrationsbereich, verwandte Links, Umfragen usw. In der Blöckeverwaltung kann entschieden werden, ob und wenn ja wo die einzelnen Blöcke angezeigt werden.

2.2.2 Newsletter

Der Newsletter des PHPNuke-Systems kann wahlweise an alle registrierten Benutzer oder nur an solche, die sich für den Newsletter gesondert eingetragen haben, geschickt werden.

2.2.3 Allgemeine Seiteneinstellungen

Bei den allgemeinen Seiteneinstellungen werden die Inhalte für Header und Footer, sowie die allgemeinen meta-Tags, d.h. z.B. der Seitename, der Name des Autors usw., festgelegt. Hier wird auch ausgewählt, welches Theme das Default-Theme des Portals sein soll.

2.2.4 Benutzerverwaltung

Die Benutzerverwaltung in PHPNuke ist sehr einfach; es gibt nur drei verschiedene Arten von Usern: Anonyme User, registrierte User und Administratoren. Alle registrierte User haben die gleichen Rechte, und sie können auch nicht in Benutzergruppen unterteilt werden. Lediglich den Administratoren kann man unterschiedliche Rechte zuweisen.

3. Erweiterungsmöglichkeiten

3.1 Mögliche Anwendungsgebiete für eigene Module

Denkbar wären zum einen Module, die die Funktionalität des PHPNuke-Systems aus Usersicht erweitern(z.B. ein Gästebuch oder ein Forum), oder solche, die die Anbindung von anderen Datenbanken ermöglichen. Wünschenswert wären z.B. auch Module zur Versionierung, zur Erstellung von CSS-Stylesheet, eine Entwicklungsumgebung für eigene Module u.ä. Besonders ehrgeizige Ansätze wären, die Blöckeverwaltung und die Benutzerverwaltung umzustrukturieren.

3.2 Erstellung eigener Themes

Zur Erstellung eigener Themes empfiehlt es sich – sofern man nicht ein Experte in PHPNuke und PHP ist – ein *bereits bestehendes Theme* auf die folgende Weise zu *modifizieren*: In der Datei `themes.php` im Hauptverzeichnis des Themes muss man alle *Namensbezeichnungen* des Themes durch den Namen des neuen Themes ersetzen. Desweiteren ändert man auch hier die *Werte für Hintergrundfarben von Elementen*, die üblicherweise in PHP-Variablen gespeichert sind. Das Theme-Hauptverzeichnis muss auch den Namen des Themes tragen. Im Ordner `images/` kann man die entsprechenden Grafikdateien mit Hilfe eines Grafikprogramms nach den eigenen Vorstellungen modifizieren. Im Ordner `styles/` kann man das CSS-Stylesheet `style.css` z.B. durch *Veränderung von Schriftarten, Farbdefinitionen* usw. verändern. Das Theme wird nach der Erstellung im Administrationsmenü unter dem Punkt für allgemeine Seiteneinstellungen aktiviert.

4. Abschließende Bewertung

4.1 Bewertung nach den Kriterien des Vortrags Content-Management

Workflow: Dadurch, dass man die relevanten Daten sehr einfach und intuitiv über das Webinterface eingeben kann, präsentiert sich hier PHPNuke als sehr komfortabel.

Freigabeverfahren: Zum einen werden Artikel von Administratoren direkt eingefügt, während Artikel von registrierten Benutzern erst nach der Zustimmung durch die Administratoren eingebunden werden. Hier wäre ein differenzierteres Rechtesystem wünschenswert.

Versionskontrolle: In PHPNuke gibt es keine Möglichkeit zur Versionskontrolle.

Unterstützung von Datenbanken: Alle gängigen SQL-Datenbanken (siehe Punkt 1.3).

Betriebssysteme: PHPNuke ist betriebssystemunabhängig.

Suchmaschine: Teilweise werden meta-Tags erzeugt. Eine automatische Eintragung in Suchmaschinen erfolgt allerdings nicht.

Auch die Forderung nach Trennung von Design, Funktionalität und Inhalt kann man in PHPNuke erkennen. So wird das Design durch Themes, die Funktionalität durch die PHP-Skripte der Module und der Inhalt durch die SQL-Datenbank geliefert. Insofern kann man bei PHPNuke von einem Web-Content-Management-System sprechen.

4.2 Persönliches Fazit

Zunächst ist es natürlich sehr vorteilhaft, dass man eine große Menge von Informationen durch verhältnismäßig wenige PHP-Skripte und eine Datenbank darstellen kann und mit einer intuitiven Benutzeroberfläche ohne Programmierkenntnisse leicht neue Seiten erstellen kann. Zudem bietet das System Erweiterungsmöglichkeiten, die allerdings vor allem im Fall der Module schon recht kompliziert und nur für erfahrene PHP-Programmierer zu empfehlen sind. Der Hauptnachteil des Systems ist das rudimentäre Rechtesystem. Dadurch, dass man keine feineren Abstufungen der Benutzerrechte vornehmen kann, können diese auch keine Arbeit an ausgewählte User delegieren. Dies bedeutet für die Administratoren mehr Arbeit, die bei sehr großen Portalen wohl nicht mehr zu bewältigen ist. Aber auch schon bei kleineren Portalen, bei denen ein hohes Maß an Interaktion erwünscht ist – wie z.B. ein Portal für Programmierer mit Foren zu Fachfragen – stellt dies ein bedeutendes Hemmnis dar. Daher empfiehlt sich PHPNuke allenfalls für kleinere bis mittelgroße Portale, die den Hauptzweck der puren Informationsdarstellung haben (z.B. Newsportale).

5. Literaturangaben

- <http://phpnuke.org/>
- <http://www.phpnuke.de/>
- <http://www.php-nuke.de>
- <http://www.phpnuke-book.com/>
- <http://www.splatt.it/phpWebNotes/EN-Book/>