Universität Paderborn
Fakultät für Elektrotechnik, Informatik und Mathematik
Warburger Straße 100
33100 Paderborn

Projektgruppe: Generierung von Webanwendungen aus visuellen Spezifikationen

Seminarausarbeitung

Existierende Systeme I: Bibliotheken & Frameworks

von
Christian Schneider

Paderborn, den 18. Juni 2004

Inhaltsverzeichnis

1 Einführung

Eine der wichtigsten auf der Kommunikationsplattform *Internet* aufbauenden Anwendungen ist das World-Wide-Web (WWW). Das WWW basiert auf der Verbindung von Ressourcen, die über das ganze Internet verstreut liegen können. Solche Ressourcen können zum Beispiel andere Dokumente, Bilder oder Audio-Dateien sein. Durch entsprechende Links in einem Web-Dokument kann man auf weitere Ressourcen verweisen. In einem solchen Link sind unter anderem der Web-Server, auf dem die Ressource liegt, und der Name der Ressource codiert. Der Client, zum Beispiel ein Web-Browser, schickt eine Nachricht an den entsprechenden Web-Server, in der er um die Zusendung der entsprechenden Ressource bittet. Der Kommunikationsablauf zwischen einem Client und einem Web-Server sowie der Aufbau von gültigen Anfragen sind im *Http-Protokoll* spezifiziert. Das Http-Protokoll ist zustandslos. Das bedeutet, dass nacheinander von einem Client an einen Web-Server gerichtete Anfragen voneinander unabhängig sind. Bei der Realisierung von verschiedenen Web-Anwendungen, zum Beispiel bei personalisierten Web-Portalen, muss die Zustandslosigkeit überwunden werden. Im späteren Verlauf der Ausarbeitung werden Techniken vorgestellt, die auf einer höheren Ebene die Zustandslosigkeit des Http-Protokolls überbrücken.
Mit der Einführung des WWW beschränkte man sich zunächst auf Dokumente mit rein statischem Inhalt. Die Dokumente lagen in Form einer Datei auf dem Web-Server vor und wurden bei entsprechenden Anfragen von Clients unverändert an diese übermittelt. Mit dem Aufkommen neuer Anwendungsfelder waren statische Dokumente unpraktikabel bzw. viele Anwendungen erst gar nicht realisierbar. Man betrachte zum Beispiel heutige Suchmaschinen im Web. Der Benutzer gibt einen Suchbegriff ein und nach kurzer Zeit wird ihm das Ergebnis in Form eines Web-Dokumentes übermittelt. Das Ablegen von statischen Dokumenten, welche die Suchergebnisse für jeden möglichen Suchbegriff und deren Kombinationen widerspiegeln, ist aufgrund des Volumens unmöglich. Als Lösung für dieses Problem bot sich die dynamische Generierung von Web-Dokumenten an. Die Dokumente werden nicht mehr statisch auf dem Web-Server abgelegt, sondern bei jeder Anfrage eines Clients dynamisch durch ein Programm erzeugt und dann an diesen übermittelt. Dies bietet zugleich die Möglichkeit, den Inhalt von Web-Seiten zu individualisieren, das heißt den Inhalt abhängig vom jeweilig anfragenden Benutzer individuell zu erstellen. Anwendungsgebiete hierfür sind zum Beispiel Web-Portale und elektronische Marktplätze. Solche Web-Anwendungen bestehen größtenteils aus dynamischen Web-Dokumenten.
Die Erzeugung von dynamischen Web-Dokumenten, zum Beispiel auf HTML basierend, erfolgt mit Hilfe von Programmen bzw. Skripten, welche die Ausgaben erzeugen, die dann mit Hilfe des Web-Servers an den Client übermittelt werden. Diese Programme laufen Server-seitig und werden bei jeder Anfrage des entsprechenden Dokumentes durch einen Client vom Web-Server ausgeführt. Als Ergebnis erhält dieser die Ausgabe des Programms. Zur Kommunikation zwi-

schen Programm und Web-Server dient die standardisierte CGI-Schnittstelle. Die Programme können in jeder beliebigen Programmiersprache bzw. Skriptsprache geschrieben werden die über die notwendige CGI-Schnittstelle verfügt. Zu Beginn wurden häufig *C* oder *Perl* verwendet. Beiden Sprachen mangelt es allerdings an Objektorientiertheit, so dass größere Projekte die daraus enstehenden Probleme aufweisen. Die Skriptsprache Perl ist zwar mächtig, aber selbst kleine Skripte können schnell komplex und dadurch schwer wartbar werden. Zwei weitere Sprachen sind die Skriptsprache *PHP* und die Programmiersprache *Java*. Beide Sprachen erfreuen sich immer größerer Beliebtheit bei der Programmierung von Web-Anwendungen mit dynamischen Inhalten.

Das Ziel dieser Ausarbeitung ist es, die Grundlagen und Konzepte beider Sprachen kurz vorzustellen sowie deren Vor- und Nachteile zu erläutern, um daraus später deren Anwendungsgebiete gegeneinander abzugrenzen. Ziel ist es nicht, eine Einführung in Web-Programmierung mit Hilfe von PHP oder Java zu geben. Deswegen sind Vorkenntnisse im Bereich Web-Programmierung und Programmiersprachen/Skriptsprachen auf jeden Fall notwendig. An den entsprechenden Stellen wird auf vertiefende Literatur hingewiesen. Desweiteren wird für PHP die Bibliothek *Pear* und für Java das Framework *Struts* vorgestellt.

2 PHP

Die Skriptsprache PHP (Personal Homepage Tools) [1] wurde 1994 ursprünglich von der Privatperson Rasmus Lerdorf entwickelt. PHP ist eine serverseitig ausgeführte Skriptsprache und dient der Erzeugung von HTML-Dokumenten mit dynamischen Inhalt. PHP-Programme werden nicht, wie zum Beispiel C-Programme, in Maschinencode übersetzt, sondern werden durch einen entsprechenden PHP-Interpreter ausgeführt. Ein großer Vorteil von PHP ist die Verfügbarkeit auf allen gängigen Plattformen, wie zum Beispiel Linux, Windows und MacOS. Der Inhalt dieser Ausarbeitung basiert auf Version 4.0 von PHP. Die Version 5.0 ist zwar bereits verfügbar, allerdings nur als Beta-Version. Die Version 5.0 weist viele Neuerungen auf, insbesondere in Bezug auf Objektorientiertheit. Deswegen wird an gegebenen Stellen darauf hingewiesen.
PHP-Programme bzw. Skripte bestehen aus normalen HTML-Dokumenten mit eingebetteten PHP-Anweisungen. Der umliegende HTML-Code ist statischer Inhalt des Dokumentes und wird unverändert übernommen bzw. an den Web-Server weitergereicht. Die PHP-Anweisungen werden durch den PHP-Interpreter ausgeführt und die dabei erzeugten Ausgaben werden an Stelle der Anweisungen in das Dokument eingefügt. Die Kommunikation mit dem Web-Server verläuft über die CGI-Schnittstelle und wird im folgenden Unterabschnitt näher erläutert.

2.1 Anbindung an den Web-Server

Die Kommunikation mit dem Web-Server erfolgt wie bereits erwähnt über die CGI-Schnittstelle. Zur Verdeutlichung der folgenden Erläuterungen sollte die Abbildung 1 herangezogen werden. Sie beschreibt die Kommunikation zwischen Client, Web-Server und PHP-Interpreter.

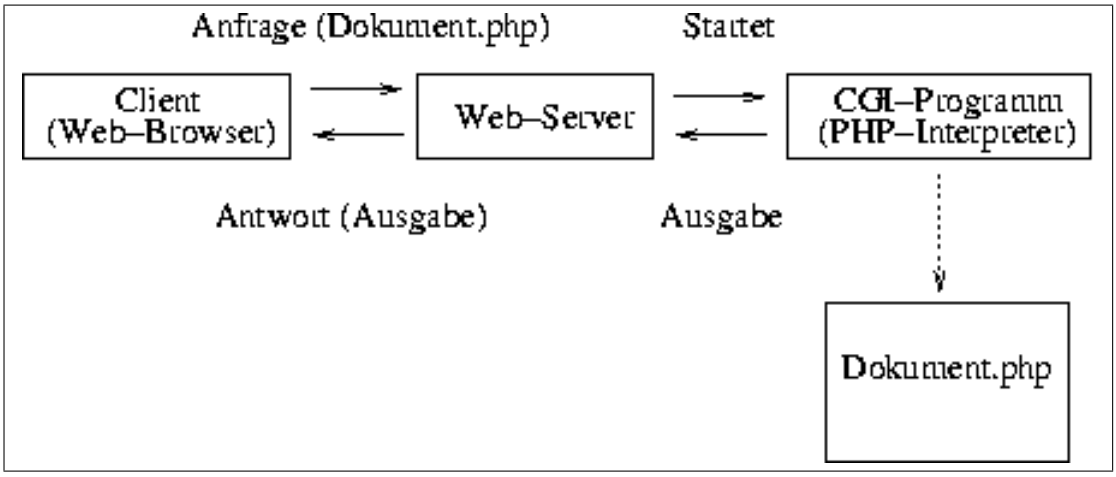

Abbildung 1: Kommunikation: Client, Web-Server und PHP-Interpreter

Ein Client (Web-Browser) fordert ein Web-Dokument (Dokument.php) mittels einer Http-Anfrage beim Web-Server an. Anhand der Dateiendung erkennt der Web-Server, dass es sich um ein PHP-Skript handelt und startet den PHP-Interpreter, um das Skript interpretieren zu lassen und die erzeugte Ausgabe an den Client weiterzuleiten.
Ein großer Nachteil der CGI-Schnittstelle ist, dass zur Bearbeitung jeder Client-Anfrage ein CGI-Programm, in diesem Fall der PHP-Interpreter, gestartet wird. Der damit verbundene *Overhead* bei der Erzeugung eines Betriebssystem-Prozesses verschlingt nicht nur Ressourcen wie Speicher, sondern nimmt auch viel Zeit in Anspruch. Die Bearbeitung anderer Client-Anfragen wird in dieser Zeit blockiert und die Bearbeitungszeit bzw. Antwortzeit des Web-Servers verlängert sich. Dadurch kann es bei zu vielen nebenläufigen Anfragen durch Speicherknappheit oder Überlastung von Rechenressourcen zu einem vollständigen Ausfall bzw. Reaktionslosigkeit des Systems kommen. Der hohe Bedarf an Rechenressourcen und die schlechte Skalierung machen PHP unattraktiv, wenn man die engere Anbindung des PHP-Interpreters an den Web-Server innerhalb eines Betriebssytem-Prozesses nutzt. In Bereichen mit einem hohen Anfrage-Aufkommen ist diese Lösung nicht vorteilhaft. Nach der Abarbeitung eines PHP-Skriptes wird dieses wieder vollständig aus dem Speicher entfernt und das CGI-Programm terminiert. Dies macht es erforderlich bei jeder Anfrage erneut das Skript in den Speicher zu laden und kostet damit wertvolle Bearbeitungszeit. Aufgrund der Beendigung

des PHP-Interpreters nach jeder Anfrage müssen in der Anwendung genutzte Datenbankanbindungen für jede Anfrage erneut zeitaufwendig geöffnet und wieder geschlossen werden.
Die Parameterübergabe zwischen Web-Server und dem PHP-Interpreter erfolgt über Umgebungsvariablen bzw. den Standardeingabekanal des PHP-Prozesses. So kann in dem PHP-Skript auf Daten zugegriffen werden, welche vom Client an den Web-Server in der Http-Anfrage mitgesendet werden. Ein Nachteil ist, dass der PHP-Interpreter nicht mit dem Web-Server kommunizieren kann, um zum Beispiel weitere Daten abzufragen. Dies bringt viele Einschränkungen für Anwendungen mit sich.

2.2 Sprache

Die Syntax von PHP ist ähnlich der von C und Java und daher für Entwickler, die aus diesem Bereich kommen, relativ schnell zu erlernen. Die aus diesen Programmierprachen bekannten Kontroll-Strukturen, wie zum Beispiel die For-Schleife und bedingte Verzweigungen sind ebenso vorhanden wie die Standard-Datentypen *int*, *boolean* und *string*. Eine Besonderheit ist, dass Variablen nicht explizit deklariert werden und untypisiert sind. Der Typ einer Variable wird zur Laufzeit vom Interpreter bestimmt und kann sich im Verlauf der Lebensdauer einer Variable ändern. Der Wert einer Variable wird bei Bedarf auf einen gültigen Typ implizit *gecastet*. Die fehlende Typsicherheit, insbesondere bei *impliziter* Typkonvertierung durch den Interpreter, kann zu unerwünschten Ergebnissen führen, denen der Programmierer nur mit einer *expliziten* Typkonvertierung entgegenwirken kann. Bei den Variablen unterscheidet man zwischen globalen, lokalen und statischen Variablen. Die sich daraus ergebenden Sichtbarkeitsbereiche und Lebensdauern von Variablen sind wie in der Programmiersprache C.
Eine weitere nützliche Fuktionalität von PHP ist die Unterstützung von regulären Ausdrücken, wie man sie von Perl kennt. Damit lassen sich flexibel Muster in Texten erkennen und sie sind damit bei der Verarbeitung von Zeichenketten ein nützliches Hilfsmittel.
Die Programmierung in PHP ist sehr stark prozedurbezogen. Es gibt zwar Klassen und Objekte, es fehlt jedoch jegliche Form der Datenkapselung. Es gibt keine Möglichkeit die Sichtbarkeit von Member-Variablen einzuschränken, wie es zum Beispiel in Java mit dem *Private*-Modifizierer möglich ist. Dieses Manko wird jedoch mit PHP Version 5.0 behoben worden sein. Desweiteren sind in der neuen Version auch Klassen-Variablen verfügbar.

2.3 Bibliotheken und Frameworks

PHP unterstützt diverse Protokolle wie zum Beispiel IMAP, Pop3, Http, ftp und Corba. Desweiteren gibt es eine strukturierte Bibliothek von Open-Source Code

für PHP. Die Pear-Bibliothek (Pear: PHP Extension and Application Repository) ist ein *Community* gesteuertes Projekt und enthält Hunderte von nützlichen Funktionen rund um die Web-Programmierung mit PHP. Ein Teilprojekt sind die PHP-Foundation-Classes (PFC). Es enthält unter anderem Klassen aus den Bereichen *Networking*, *Caching* und *Encryption*. Weitere Informationen findet man auf den Web-Seiten des Pear-Projektes [2].
Desweiteren besitzt PHP eine gute Datenbankunterstützung. Es werden alle aktuellen Datenbanken, wie zum Beispiel MySQL und Oracle unterstützt.

2.4 Session-Verwaltung

Wie bereits in der Einführung erwähnt, ist das Http-Protokoll zustandslos. Für viele Anwendungsbereiche ist es aber notwendig die anfragenden Clients unterscheiden zu können, um dadurch dem Benutzer individuell seine Dienste anbieten zu können. Als Beispiel wäre zum Beispiel eine Online-Auktion zu nennen. Damit ein Benutzer mitbieten kann, muss er sich zuerst der Web-Anwendung gegenüber identifizieren und kann dann erst seine Gebote abgeben. Im einfachsten Fall besitzt die Web-Anwendung zwei Zustände: „Benutzer ist identifiziert“ und “Benutzer ist nicht identifiziert“. Befindet sich der Benutzer im zweiten Zustand stellt die Web-Anwendung dem Benutzer mehr Funktionalität zur Verfügung, zum Beispiel das Abgeben von Geboten und das Bezahlen der Ware.

Um die Zustandslosigkeit des Http-Protokolls zu überwinden, verfügt PHP bereits über sogenannte *Sessions* und nimmt dem Web-Programmierer dadurch lästige Arbeit ab. Sessions dienen dazu Informationen über den Client für eine begrenzte Dauer auf dem Web-Server abzuspeichern. Um eine Zuordnung zwischen Client und Session zu haben, wird bei der ersten Anfrage des Clients eine *Session-Id* erzeugt. Diese wird bei jeder weiteren Anfrage vom Client an den Web-Server mitgesendet. Dadurch können die Informationen in der Session über mehrere Anfragen hinweg genutzt und aktualisiert werden. Die eingebaute Session-Verwaltung arbeitet intern mit einem Cookie oder ersatzweise mit einem zusätzlichen Http-Abfragedatum (URL-Rewriting), das per Get-Methode übermittelt wird. Bei der zweiten Methode wird die Session-Id in der Regel transparent in URLs von Web-Dokumenten der Anwendung eingebunden.

2.5 Anwendungsgebiete

Mit PHP kann man schnell einfache Web-Anwendungen entwickeln. Die Sprache PHP selbst ist einfach, aber mächtig. Weiterhin spricht die hohe Verbreitung und die hohe Unterstützung durch Internet-Service-Providern (ISP) für PHP.
Ein großer Nachteil ist die notwendige Einbettung von PHP-Anweisungen in HTML-Dokumenten. Dies macht eine saubere Trennung zwischen Anwendungs-

code und Darstellung unmöglich und reduziert deren Wiederverwendbarkeit. Somit sind Web-Anwendungen mit hohem algorithmischen Anteil weniger das Anwendungsfeld von PHP.
Auch durch die Anbindung an den Web-Server als CGI-Programm entsteht ein großer Overhead, der gerade bei hohem Anfrage-Aufkommen das System schnell überlasten kann. Die schlechte Skalierung macht den Einsatz von PHP in Bereichen wie den *eCommerce* unattraktiv.
Die geringe Objektorientiertheit und fehlende Datenkapselung verlangt dem Programmierer viel Disziplin ab, um sauberen und halbwegs wiederverwendbaren Programmcode zu entwickeln.
Somit liegt das Einsatzgebiet von PHP insbesondere im Rapid-Prototyping und der schnellen Entwicklung von kleinen bis mittelgroßen Web-Anwendungen mit geringem Anwendungscode.

3 Java

Die Programmiersprache Java wurde Anfang der 90er von der Firma Sun [3] entwickelt. Im Gegensatz zu PHP ist Java eine vollwertige und objektorientierte Programmiersprache. Java gehört ebenfalls zu den interpretierten Sprachen. Programme, die in Java geschrieben sind, werden zunächst durch den Java-Compiler in Java-Byte-Code übersetzt und können danach durch die Java-Virtual-Maschine (JVM), den Java-Interpreter, ausgeführt werden. Dadurch sind Java-Programme auf allen Plattformen ausführbar, auf denen eine JVM zur Verfügung steht.
Neben der Programmiersprache Java werden von Sun vor allem Technologien entwickelt, welche auf Java aufbauen bzw. diese nutzen. Insbesondere für den Bereich Web-Anwendungen wurden zahlreiche Technologien entwickelt, welche die Entwicklung von Web-Anwendungen erleichtern bzw. auf ein höheres Entwicklungsniveau bringen. Dies sind: *Java-Servlets*, *Java-Server-Pages (JSP)* und *Java-Beans*. Auf deren Funktionsweise und Nutzen für den Entwickler von Web-Anwendungen wird in den folgenden Unterabschnitten eingegangen. Weiteres zum Thema Java findet man in dem Buch *Handbuch der Java-Programmierung* [5] von Guido Krüger.

3.1 Servlets

Servlets sind Java-Programme, die auf der Server-Seite ausgeführt werden und zur Bearbeitung von Client-Anfragen dienen. Sie stehen dem Programmierer über die Java-Klassenbibliothek zur Verfügung. Durch das Ableiten von der Klasse *Servlet* kann man diese um anwendungsspezifische Funktionalität erweitern.
Für die Ausführung von Servlets benötigt man eine Servlet-Engine. Einfach ausgedrückt ist dies eine spezielle JVM die zusätzlich über eine Web-Server-Anbindung verfügt. Die Servlet-Engine erhält eingehende Nachichten vom Web-

Server und führt die dazu gehörigen Servlets aus, indem sie die entsprechenden Methoden des Servlets mit den notwendigen Parametern aufruft. Das Programm *Tomcat* [4] von der *Apache Software Foundation* ist eine solche, voll funktionsfähige und zudem frei verfügbare Servlet-Engine.
Das Web-Dokument, welches erzeugt werden soll, wird durch das Schreiben von Text-Fragmenten, zum Beispiel HTML-Anweisungen, in einen Ausgabe-Strom aufgebaut. Damit sind bei Servlets der statische Teil des Dokumentes in Java-Code eingebettet und nicht umgekehrt wie bei PHP. Somit hat man die Sicht eines Programmierers und nicht die eines Web-Designers, der PHP-Anweisungen in HTML einbettet. Dies ist inbesondere bei Web-Dokumenten von Vorteil, die viel Anwendungscode erfordern und damit die Java-Anweisungen überwiegen. In den Fällen, wo der statische Teil eines Web-Dokumentes überwiegt, ist die Verwendung von Servlets unnatürlich und nicht performant. Zusätzlich sind für die Programmierung von Servlets Programmierkenntnisse notwendig und sie sind daher in erster Linie ein Werkzeug für Programmierer und nicht für Web-Designer. Dies haben auch die Entwickler von Sun erkannt und die Java-Server-Page Technologie entwickelt, die im nächsten Unterabschnitt vorgestellt wird.

3.2 Java-Server-Pages (JSP)

Bei Java-Server-Pages ist der Java-Code wie bei PHP zwischen speziellen Tags in HTML-Code eingebettet. Dabei dient der HTML-Code für den statischen Teil und die eingebetteten Java-Anweisungen für den dynamischen Teil und bieten damit eine einfache Möglichkeit, statischen und dynamischen Inhalt zu mischen. Das beste Einsatzgebiet für JSP ist die Darstellungsschicht, also dort wo wenig Anwendungscode erforderlich ist und der statische Teil überwiegt. Aus technischer Sicht sind JSP eine spezielle Variante von Servlets, da sie transparent für den Entwickler bei der ersten Anfrage automatisch in Servlets übersetzt werden. Sie sollen in erster Linie die Entwicklung der Darstellungsschicht natürlicher Gestalten und sind somit ein Werkzeug für Web-Designer. Nützlich ist auch, dass man innerhalb von JSP auf Servlets oder Java-Beans zugreifen kann. So kann man die komplexen Teile der Anwendung mit Hilfe von Servlets implementieren und die Darstellung in den JSP belassen. Dadurch kann man eine klare Trennung zwischen Anwendungscode und Darstellung, sowie Programmierung und Web-Design erzielen. Diese klare Trennung fehlt in PHP. Mehr zu dem Thema Java-Server-Pages findet man im gleichnamigen Buch [6] von Volker Turau.

3.3 Java-Beans

Java-Beans sind eigenständige Softwarekomponenten und bieten ihre Funktionalität nicht direkt dem Benutzer in Form einer Anwendung an. Sie stellen vielmehr ihre Funktionalität anderen Anwendungen zur Verfügung, um so deren Funktionalität auf einfache Art und Weise zu erweitern, ähnlich wie bei einer Funktions-

oder Klassenbibliothek. Java-Beans sind besondere Software-Komponenten, bei denen die Informationen über die verfügbare Funktionalität in der Komponente selbst steckt. Diese Information über vorhandene Methoden und Attribute lassen sich zur Laufzeit über einen standardisierten Zugriffsmechanismus erfragen. Ein Ziel bei Java-Beans und anderen Komponentenmodellen liegt darin, einen hohen Grad an Wiederverwendbarkeit des Programmcodes zu erreichen.
Im Bereich der Web-Anwendungen dienen Java-Beans dem gleichen Zweck wie Servlets, also der Implementierung von Anwendungscode. Allerdings haben Java-Beans gegenüber Servlets den Vorteil, dass sie nicht auf Web-Anwendungen beschränkt sind. Dadurch lässt sich der Anwendungscode auch in vom Web unabhängigen Anwendungsprogrammen verwenden. Desweiteren können sie ihre Funktionalität aber auch in Form von Web-Services anbieten und haben damit ein breiteres Anwendungsfeld als Servlets.
Wie Servlets werden Java-Beans bei der ersten Anfrage eines Clients persistent in den Speicher geladen, wodurch jede weitere Anfrage schneller bearbeitet werden kann. Zusätzlich sind nebenläufige Zugriffe durch mehrere Clients möglich. Aus diesem Grund müssen Java-Beans *thread-safe* programmiert werden.

3.4 Anbindung an den Web-Server

Im Gegensatz zu PHP kommuniziert die Servlet-Engine nicht über die standardisierte CGI-Schnittstelle mit dem Web-Server, um die bereits erläuterten Nachteile von CGI zu vermeiden. Zur Verdeutlichung der folgenden Erläuterungen kann die Abbildung 2 herangezogen werden. Sie beschreibt die Kommunikation zwischen Client, Web-Server und Servlet-Engine.

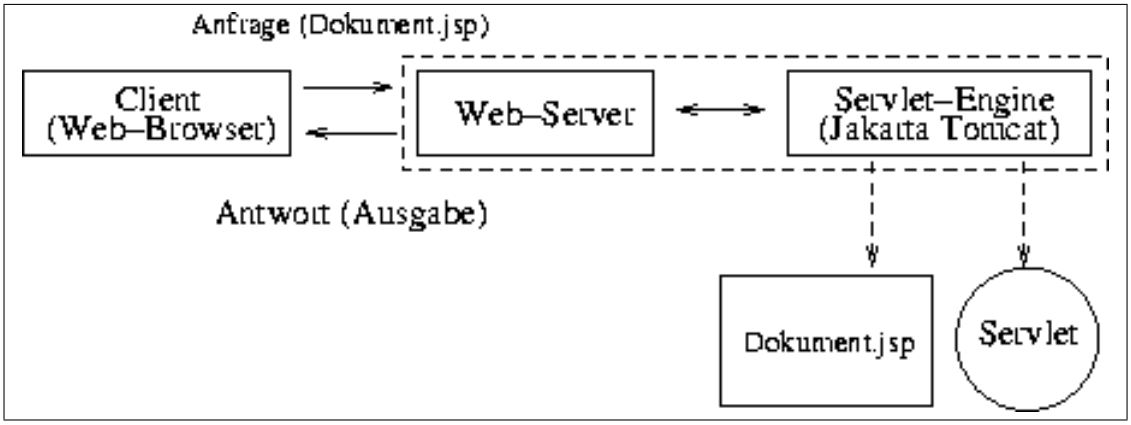

Abbildung 2: Kommunikation: Client, Web-Server und Servlet-Engine

Ein Client fordert ein Web-Dokument (Dokument.jsp) mittels einer Http-Anfrage an. Die Anfrage wird aufgrund der Dateiendung an die Servlet-Engine, zum Beispiel Jakarta Tomcat, weitergereicht. Diese schaut nach, ob das zu diesem Do-

kument gehörige Servlet bereits vorliegt. Ist dies nicht der Fall, muss es sich um die erste Anfrage nach diesem Dokument handeln, wird nach dem Quelltext der JSP gesucht und diese in ein Servlet übersetzt. Das gleiche geschieht auch dann, wenn das JSP-Dokument neuer als das Servlet ist, das JSP-Dokument also in der Zwischenzeit modifiziert worden ist. Steht das Servlet zur Verfügung, lädt die Servlet-Engine das Servlet *persistent* in den Speicher. Das Servlet verbleibt also nach der Bearbeitung der Anfrage im Speicher, anders als das Verhalten bei PHP. Dadurch kann jede weitere Anfrage noch schneller beantwortet werden. Ein wesentlicher Nachteil dieser Technik ist der höhere Speicherverbrauch. Danach erzeugt die Servlet-Engine eine Instanz der geladenen Servlet-Klasse und ruft die entsprechenden Methoden auf. Dabei wird die Http-Anfrage mit den enthaltenen Daten in ein *HttpRequest*-Objekt verpackt und dem Servlet zur weiteren Nutzung mit übergeben. Nach der Bearbeitung wird das Servlet-Objekt wieder zerstört.
Die Servlet-Engine wird, wie der Web-Server, nur einmal zu Beginn gestartet und es existiert nur eine Instanz. Dies ist ein wesentlich effizienteres Verfahren, als es bei CGI-Programmen bzw. PHP der Fall ist.
Für die nebenläufige Verarbeitung von Anfragen werden Java-Threads verwendet und nicht Threads bzw. Prozesse des Betriebssystems. Diese können schneller erzeugt werden und der Aufruf von blockierenden Betriebssystem-Funktionen wird vermieden. Dadurch besitzt Java eine gute Skalierung und eignet sich bei Anwendungen mit einem hohen Anfrage-Aufkommen besser als PHP.
Zusätzlich verfügt die Servlet-Engine über sogenannte *Connection-Pools*. In diesen werden einmal angefordete Datenbankverbindungen zwecks Wiederverwendung persistent gehalten. Versucht ein Servlet eine Datenbankverbindung zu öffnen und es befindet sich noch mindestens eine im Pool, dann wird das Servlet aus diesem bedient. Dadurch wird eine bessere Performance bei Anwendungen mit Datenbank-Nutzung erzielt.

3.5 Sprache

Im Gegensatz zu PHP ist Java eine vollwertige Programmiersprache und nicht auf den Einsatz in Web-Anwendungen beschränkt. Die Vollwertigkeit macht sich insbesondere im Bereich der Objektorientierung bemerkbar. Zu nennen wären da Vererbung, Datenkapselung, Schnittstellen und Polymorphismus. Dies sind nützliche Hilfsmittel auf Sprachebene für die Realisierung höherer Software-Paradigmen wie sie zum Beispiel von diversen Design-Patterns bekannt sind. Durch deren Anwendung wird der Programmcode verständlicher, flexibler, wartbarer und verbessert letztendlich damit auch dessen Wiederverwendbarkeit.
Jedoch verfügt Java in der Kernsprache über keine regulären Ausdrücke, sondern ist auf die ausgiebige Nutzung von Bibliotheken angewiesen.

3.6 Bibliotheken und Frameworks

Bei der Entwicklung mit JSP, Servlets und Java-Beans kann man die ganze Funktionalität der Java-Klassen-Bibliothek nutzen (JFC). Es gibt keine Einschränkungen. So können unter anderem auch Technologien wie RMI, Sockets, Imaging und Threads eingesetzt werden. Durch die Verwendung von Threads kann der Entwickler selbst Teile der Anwendung nebenläufig ablaufen lassen, was insbesondere bei aufwendigen Berechnungen sinnvoll ist. Neben weiterer Funktionalität in Form der Java-Klassenbibliothek gibt es zahlreiche auf den Java-Technologien basierende Frameworks, die auf die Entwicklung von Web-Anwendungen zugeschnitten sind. Frameworks bieten Funktionalität auf einer höheren *problemspezifischen* Ebene, im Gegensatz zu reinen Funktions- oder Klassenbibliotheken. Eines dieser Frameworks ist *Struts* [8] von der *Apache Software Foundation*.

Struts ist ein Open-Source-Framework für Web-Anwendungen basierend auf Java. Entworfen wurde es, um leichten Zugriff auf den Anwendungscode der Web-Anwendung zu haben. Die Software-Architektur von Struts beruht auf dem *Model 2*, dies ist eine Variation des bekannten Software-Paradigmas *Model-View-Controller*. Das *Modell* enthält und verwaltet die notwendigen Daten, das *View* ist für die Darstellung der Daten verantwortlich und der *Controller* verarbeitet Eingaben und enthält die Ablaufsteuerung. Dabei verwendet der Controller das Modell zur Speicherung der Daten und den *View* für deren Darstellung. Der Kern von Struts besteht aus einer flexiblen Kontrollschicht, wodurch sich das Framework leicht für eigene Anwendungen erweitern lässt.
Die Haupt-Controller-Komponente wird im Struts-Framework durch ein Objekt vom Typ *ActionServlet* repräsentiert. Dieses Objekt verwaltet wiederum eine Menge von sogenannten *ActionMappings*. Zur Verdeutlichung kann Abbildung 3 herangezogen werden.

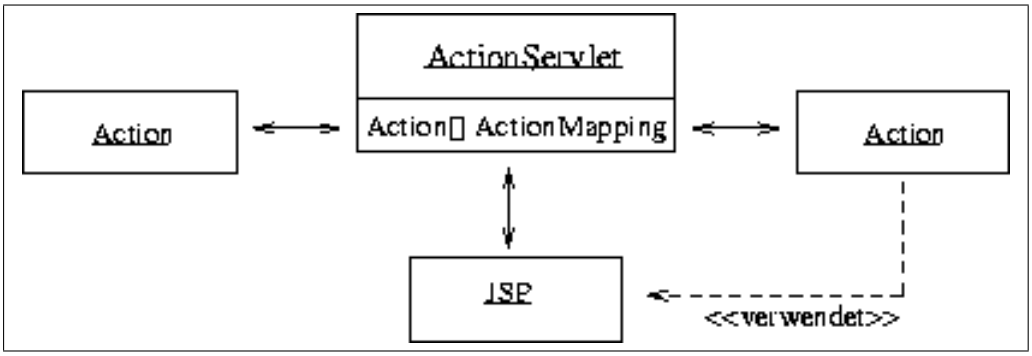

Abbildung 3: Struts-Framework

Der Schlüssel bei einem ActionMapping definiert einen Pfad der mit einer URL-

Anfrage übereinstimmt. Der voll qualifizierte Name einer von der *Action-Klasse* abgeleiteten Klasse ist der zu einer URL-Anfrage gehörende Wert. Dadurch wird eine Zuordnung von Anfrage-URLs zu verantwortlichen Bearbeitungsklassen realisiert. Die Klassen basieren auf Servlets und dienen als spezialisierte Request-Handler. Bei einer Anfrage ermittelt das ActionServlet-Objekt mit Hilfe des ActionMappings die verantwortliche Klasse, erzeugt ein Objekt von dieser Klasse und delegiert die Anfrage an das Objekt weiter. Die Request-Handler enthalten den anwendungsspezifischen Anwendungscode und übernehmen das Antworten auf Client-Anfragen. Das ActionMapping erlaubt die einfache Abbildung von Http-Anfragen auf Anwendungscode. Desweiteren kann ein Request-Handler mit Hilfe der Controller-Komponente Anfragen weiterleiten, zum Beispiel an eine Java-Server-Page, welche die Darstellung übernimmt. Für die Einbettung von dynamischem Inhalt in JSP enthält Struts eine umfangreiche Tag-Bibliothek.
Die Hauptaufgabe des Benutzers von Struts liegt bei der Programmierung von Request-Handlern für jede logische Anfrage, die empfangen werden kann, sowie der Konfiguration des ActionMappings. Struts ist gedacht für größere Projekte mit einer klaren Trennung von Anwendungscode und Darstellung.

3.7 Session-Verwaltung

Auch im Java-Framework gibt es eine eingebaute Session-Verwaltung. Zu diesem Zweck gibt es die Klasse *HttpSession* in der Java-Klassen-Bibliothek (JFC). Bei der Erzeugung einer Instanz der Klasse HttpSession wird zuerst geprüft, ob bereits eine Session für diesen Client existiert, wenn ja, wird diese verwendet, andernfalls wird eine neue erzeugt. Um den Client bei der nächsten Anfrage wieder zu erkennen, wird diesem die Session-Id übermittelt. Diese wird bei weiteren Anfragen des Clients mittels URL-Rewriting an den Web-Server mitgesendet. Diese Technik entspricht der zweiten Alternative zur Realisierung von Sessions in PHP.

3.8 Anwendungsgebiete

Die auf Basis von Java entwickelten Web-Technologien wie JSP, Servlets und Java-Beans dienen nicht primär dazu, schnell Software zu entwickeln, sondern vielmehr als Basis für die Entwicklung komplexer Web-Anwendungen, die flexibel, wartbar und wiederverwendbar sein sollen. Für Web-Anwendungen mit einer hohen Lebensdauer. Die Möglichkeit der klaren Trennung von Anwendungscode und Darstellung fördert die Wartbarkeit und Wiederverwendbarkeit von Komponenten. Aufgrund der guten Skalierungs-Eigenschaften und der Sicherheitsmechanismen eignet sich Java mit den bereits vorhandenen Web-Technologien besonders für die Entwicklung komplexer Web-Anwendungen mit einem hohen Anteil an Anwendungscode.

Ein wesentlicher Nachteil ist allerdings die noch geringe Unterstützung durch Internet-Service-Providern.

4 Fazit

Die Programmiersprache Java und deren speziell entwickelten Technologien für Web-Anwendungen bilden zusammen eine komplette Software-Architektur zur Realisierung von komplexen Web-Anwendungen. Dabei steht nicht die schnelle Entwicklung im Vordergrund. Mit der Objektorientiertheit von Java lassen sich einfach höhere Software-Paradigmen verwirklichen, um strukturierte, wartbare und wiederverwendbare Software zu schreiben. Desweiteren besitzt Java eine gute Skalierung und findet daher insbesondere im eCommerce Umfeld seinen Einsatz. Die auf Java-Technologien basierenden Frameworks, wie zum Beispiel Struts, nehmen dem Entwickler durch einen vorgefertigten aber flexiblen Rahmenalgorithmus und viele zusätzliche Hilfsfunktionen eine Menge Arbeit ab. Dadurch kann dieser sich voll auf den anwendungsspezifischen Teil bei der Entwicklung von Web-Anwendungen konzentrieren und muss sich nicht mit technischen Details auseinander setzen.
Im Gegensatz zu Java eignet sich PHP eher für die schnelle Entwicklung von Web-Anwendungen im Kleinen. Einfache Dinge lassen sich leicht realisieren und die Pear-Bibliothek bietet eine Vielzahl an nützlichen Funktionen aus den verschiedensten Bereichen. Durch die noch fehlende Objektorientiertheit von PHP lassen sich viele Konzepte nicht realisieren. Dies schränkt die Abstraktionsmöglichkeiten der Pear-Bibliothek ein. Eine weitere Folge daraus ist auch der Mangel an fertigen Frameworks. Durch die schlechtere Skalierung ist PHP nur bedingt für Web-Anwendungen mit einem hohen Anfrage-Aufkommen geeignet. Trotz dieser Einschränkungen kann PHP eine hohe Verbreitung aufweisen.
Java oder PHP? Letztendlich ist Typ und Umfang der zu entwickelnden Web-Anwendung für die Wahl entscheidend. Dabei kann diese Ausarbeitung als Basis für eine leichtere Entscheidungsfindung dienen.

Literatur

[1] PHP-Homepage: *PHP: Hypertext Preprocessor* Online: http://www.php.net

[2] Pear-Homepage: *PEAR - PHP Extension and Application Repository* Online http://pear.php.net/

[3] Sun-Homepage: *Java Technology* Online: http://java.sun.com/

[4] Tomcat-Homepage: *The Jakarta Site - Apache Tomcat* Online: http://jakarta.apache.org/tomcat/

[5] Guido Krüger: *Handbuch der Java-Programmierung, 3.Auflage*, Addision-Wesley, 2003

[6] Volker Turau: *Java Server Pages -Dynamische Generierung von Web-Dokumenten-*, dpunkt.verlag, 2000

[7] Java-Web-Services: *The Java Web Services Tutorial 1.0* Online: http://java.sun.com/webservices/docs/1.0/tutorial/

[8] Struts-Homepage: *The Apache Struts Web Application Framework* Online: http://jakarta.apache.org/struts/