

Vorwort

Der zunehmend differenzierten Anwendung von Web-Technologien steht ein krasser Mangel an grundlegender Literatur gegenüber. Es ist ein Anliegen dieses Buches, Softwareentwicklern einen profunden Einblick in alle relevanten Themenbereiche zu geben. Der Leser wird am Browser abgeholt und durch die dominierenden Technologien auf der Client- und der Serverseite geführt. Dort angekommen, werden die entstandenen Lösungsansätze auf Architekturfragen hin untersucht und die Anbindung bestehender Systeme in das Gesamtbild eingefügt. Auf dem beschriebenen Weg werden die für Web-Technologien grundlegenden Konzepte vorgestellt und die unterschiedlichen Programmiermodelle anhand vieler Beispiele praxisnah illustriert.

Web-Technologien haben sich zu einer festen IT-Disziplin entwickelt. Was vor 10 Jahren in öffentlichen Instituten und privaten Garagen begann, genießt heute strategische Bedeutung. Die für Web-Applikationen eingesetzten Technologien haben sich im Gleichschritt mit den gestiegenen Anforderungen professionalisiert. Web-Applikationen sind zu einem Top-Thema der IT geworden. Sie tragen in vielen Anwendungsfeldern zur Wertschöpfung bei: von der Kommunikation im Unternehmen über Transaktionen mit Kunden und Partnern bis zur Integration komplexer bestehender Anwendungen.

»Web-Technologien« bietet einen kompakten Einstieg in die vielfältigen Aspekte moderner Web-Applikationen. Das Buch ist keine lose Aneinanderreihung isoliert zu betrachtender Technologien. Eingebettet in übergeordnete sachliche Zusammenhänge veranschaulichen die Technologien vielmehr die fundamentalen Problemstellungen von Software der Gattung Web-Applikation. Eine Vielzahl von Beispielen unterstützen den Leser dabei, Anwendungsmöglichkeiten, Techniken und typische Lösungsansätze besser zu verstehen. Die Beispiele werden jeweils in der für eine Technologie üblichen Programmiersprache präsentiert. Für das Verständnis der Beispiele wurde zu XML, CSS, XSL, JavaScript, Java/C#, Perl und PHP jeweils eine kurze Einführung in Syntax und Semantik eingearbeitet. Neben vielen unabhängigen Beispielen demonstriert ein immer wieder implementierter einfacher Bestellprozess das Zusammenspiel komplementärer sowie die Stärken und Grenzen konkurrierender Technologien auf.

Für wen dieses Buch geschrieben ist

Dieses Buch richtet sich an Softwareentwickler, Ingenieure und alle Interessierten mit Erfahrung in der Softwareentwicklung. Der Leser sollte mit der prozeduralen Programmierung durch praktische Erfahrung vertraut sein und die Konzepte der objektorientierten Programmierung kennen und verstehen. Darüber hinaus ist aufgrund der Vielzahl von Programmiersprachen und Technologien ein gewisses Abstraktionsvermögen für den einen oder anderen Quellcode notwendig. Hyperlinks auf vertiefende Tutorials zu den einzelnen Technologien befinden sich auf der begleitenden Website. Der Leser sollte darüber hinaus über grundlegende HTML-Kenntnisse verfügen. Erfahrungen mit anderen Web-Technologien sind sehr hilfreich, aber nicht notwendige Voraussetzung.

Website

Unter <http://www.woehr.com/Web-Technologien> wurde eine begleitende Website eingerichtet. Neben den Buch-Beispielen finden Sie dort:

- Links auf Tutorials zu allen im Buch thematisierten Programmiersprachen und Frameworks
- Links zu allen relevanten Standards
- Links auf viele Websites mit weiterführenden Informationen zu den behandelten Themen

Organisation der Inhalte

Teil I beleuchtet das *Internet und das World Wide Web* aus unterschiedlichen Perspektiven. Nach einer Einordnung der zehn zentralen technischen und konzeptionellen Begriffe wird das *dynamische Web* vorgestellt. Dynamische Dokumente werden von dynamisch generierten Dokumenten abgegrenzt. Es wird aufgezeigt, welche Technologien für Interaktivität im Browser sorgen und wie serverseitige Programme aktuelle Inhalte generieren. Das *angewandte Web* fragt nach dem Anwendungsumfeld von Web-Applikationen. Aus der Charakterisierung von Intranet, Extranet und Internet werden Anwendungsdomänen erschlossen und Problemfelder identifiziert. Grundlegende Anwendungskonzepte zu deren Lösung sowie die wesentlichen Softwaregattungen werden vorgestellt.

Teil II befasst sich mit *Client-Technologien*. Nach Vorstellung von *HTML* und *XML* zur Repräsentation der Dokumentenstruktur wird das Document Object Model und dessen integrative Bedeutung für alle Client-Technologien beschrieben. Deren Zusammenspiel wird durch die Verarbeitungsschritte des Layout-Prozesses verdeutlicht. Mit *Cascading Stylesheets* und *XSL* folgen Technologien zur Präsentation von HTML und XML und damit eine Diskussion über die Trennung von Struktur und Präsentation. Schwerpunkte bilden die zentrale Konfiguration von Web-Präsentationen und die Transformation von XML nach

XHTML. Nach einem Überblick über die wichtigsten Leistungsmerkmale von *JavaScript* werden dessen Anwendungsmöglichkeiten ausführlich dargestellt. Zusätzliche Browser-Fenster werden geöffnet und dynamisch mit HTML-Inhalten gefüllt. Benutzerereignisse werden mit Event Handlern verarbeitet und in das Dokument integrierte externe Komponenten gesteuert. Nach der Einordnung von DHTML folgt ein *Java*-Exkurs. Im Mittelpunkt stehen dessen besondere und aus Sicht traditioneller Programmiersprachen neuen Leistungsmerkmale. Mit *Java Applets* wird der Komponenten-Begriff eingeführt. Nach Erörterung des Lebenszyklus wird aufgezeigt, wie grafische Oberflächen implementiert und GUI-Ereignisse verarbeitet werden können. Verschiedene Varianten für die Kommunikation zwischen Applets und serverseitigen Programmen werden aufgezeigt. Den Abschluss von Teil II bildet die zusammenfassende Beurteilung der Client-Technologien.

Teil III thematisiert *Server-Technologien*. Im Unterschied zu den sich ergänzenden Client-Technologien verfolgen serverseitige Technologien dasselbe Ziel. Sie unterscheiden sich hinsichtlich Methode, Leistungsumfang und Programmiermodell. Nach der Vorstellung klassischer Server Side Includes werden die Web-Protokolle HTTP und HTTPS erläutert. Den Mittelpunkt der Einführung bilden Sessions. Deren Bedeutung und die zur Implementierung verfügbaren Alternativen werden diskutiert. Mit *Perl* folgt ein Exkurs in eine sehr leistungsfähige Skriptsprache. Programme für das Common Gateway Interface werden mehrheitlich mit Perl implementiert und auch PHP ist daraus hervorgegangen. Im Mittelpunkt von CGI steht der programmseitige Umgang mit HTTP auf niedriger Ebene. Aufgezeigt wird auch die Kombination von clientseitiger mit serverseitiger Eingabevalidierung. Mit *PHP* wird eine Technologie vorgestellt, die wegen ihrer Einfachheit sehr beliebt ist. Dass der zugrunde liegende dokumentenzentrierte Ansatz auch bei komplexeren Web-Applikationen handhabbar bleiben kann, wird an einem umfangreicheren Beispiel demonstriert. Das hier eingeführte Shop-Beispiel »vino veritas« wird zur Veranschaulichung der unterschiedlichen Programmiermodelle in den weiteren Kapiteln noch einige Male implementiert. Mit *Java Servlets und JSP* folgt die Erläuterung der im professionellen Umfeld erfolgreichsten Technologie. Die zugrunde liegende Model-View-Controller-Architektur wird diskutiert und deren Rollenmodell am Beispiel demonstriert. *Active Server Pages* komplettieren die serverseitigen Präsentationstechnologien. Ein zusammenfassender *Vergleich* der *Server-Plattformen* rundet Teil III ab.

Teil IV diskutiert Fragen rund um die *Architektur* von Web-Applikationen. Eine Schichtenarchitektur wird vorgestellt und die Verantwortlichkeiten jeder einzelnen Schicht beschrieben, um anschließend die verschiedenen Lösungsansätze aus Teil III einer kritischen Würdigung zu unterziehen. Darauf folgt eine allgemeine Darstellung der Softwaregattung *Applikationsserver*. Es wird begründet, warum die Anforderungen an moderne Informationssysteme durch den Einsatz verteilter Komponenten in einer Internet-zentrierten IT-Welt besonders effizient

gelöst werden. Die zugrunde liegende Kommunikations-Infrastruktur wird genauso betrachtet wie das Komponentenmanagement und die typischerweise verfügbaren Systemdienste. Mit der *Java 2 Enterprise Edition* (J2EE) folgt die konkrete Spezifikation einer Applikationsserver-Plattform. Neben der Vorstellung von Enterprise JavaBeans werden die technologischen Optionen für die Anbindung bestehender Systeme diskutiert. Die Beschreibung von *.NET* beginnt mit einer vergleichenden Darstellung der Plattform-Grundlagen. Den Schwerpunkt bildet mit ASP.NET das Framework für Web-Applikationen. Dessen ereignisgesteuertes Programmiermodell verspricht eine besonders elegante Umsetzung der Model-View-Controller-Architektur. Dem *.NET*-Applikationsserver ist ein weiterer Abschnitt gewidmet. Den Abschluss bilden *Web Services*. Mit dem applikationszentrierten Web wird zunächst die Grundidee beschrieben, um danach zu begründen, weshalb Web Services eine universelle Lösung für Integrationsfragen in Aussicht stellen. Die Bausteine und Standards bilden den Ausgangspunkt für die Darstellung einer Basisarchitektur. Auf die einführende Vorstellung der zugrunde liegenden Technologien folgen Beispiele mit Java und *.NET*.

Danke

Dem Team des dpunkt.verlages möchte ich für die große Flexibilität und umfangreiche Unterstützung bei der Erstellung des Buches danken. Ein ganz besonderer Dank gilt Patrick Sbrzesny, dessen Begeisterung für *.NET* immer wieder zu neuen Einsichten geführt hat, die in das Buch einfließen konnten. Nicht vergessen möchte ich die beiden Reviewer, deren umfangreiche Vorschläge die Qualität substantiell verbessern halfen. Schließlich danke ich Gerd Siegmund, der mich dazu ermuntert hat, dieses Buch zu schreiben und gleichzeitig verschwieg, welche Mühe das macht.