

Michael Wissen

Eine komponentenorientierte Vorgehensweise zur Modellierung von Websites

A Component-Oriented Methodology for Modeling Websites

Web Engineering_Modellierung_Navigation_Website

Zusammenfassung. Die Entwicklung webbasierter Informationssysteme ist in ihrer Komplexität gleichbedeutend mit dem Entwurf und der Umsetzung konventioneller Software. Dennoch finden Ansätze des Software Engineering in der Entwicklungspraxis bislang keine Berücksichtigung, d. h. ein systematisches Vorgehen unter Einbeziehung von Modellierungsaspekten wird dem Entwicklungsprozess webbasierter Anwendungen nicht zugrunde gelegt. Dies hat nicht nur nachteilige Auswirkungen auf die Qualität der Anwendung, sondern führt aufgrund der hohen Änderungsdynamik webbasierter Anwendungen sowie komplexer Navigationsbeziehungen und Inhaltsstrukturen vor allem zu hohen Pflege- und Wartungsaufwänden. Der vorliegende Beitrag thematisiert den Entwicklungsprozess webbasierter Anwendungen, der in Analogie zu Ansätzen des Software Engineering in unterschiedliche Entwurfsbereiche unterteilt werden soll. Zu den einzelnen Bearbeitungsphasen werden verschiedene Entwurfsmodelle vorgestellt, die auf der Grundlage einer Informations- und Metadatenstruktur den Aufbau eines Navigations-, Sichten- und Präsentationsmodells einer Website unterstützen.

Summary. The development of web-based information systems is in its complexity tantamount to the design and implementation of conventional software. However, currently software engineering approaches are not applied in development practice, i.e., the development process of web-based applications is not based on a systematic process comprising modelling aspects. This has not only adverse effects on the quality of the application but, above all, results in high maintenance and support expenses due to the high modification dynamics of web-based applications as well as complex navigation relations and content structures. The following article discusses the development process of web-based applications, which is divided into several design phases similar to software engineering approaches. To each phase various design models will be introduced, supporting the development of a navigation, view and representation model.

1. Einführung

Seit der Entstehung des World Wide Web (WWW) zu Beginn der neunziger Jahre hat sich der Stellenwert webbasierter Anwendungen kontinuierlich erhöht. Durch die stetig ansteigende Zahl von Benutzern und dem dadurch verstärkten wirtschaftlichen Interesse dient das WWW inzwischen nicht nur als reines Informationssystem, sondern wird zusätzlich als Plattform kommerzieller Anwendungen genutzt. Innerhalb weniger Jahre sind neue verteilte Web-Anwendungen mit verstärkt funktionalen Merkmalen entstanden, die sich zuneh-

mend durch verteilte Prozesse, heterogene Systeme, komplexe Benutzerschnittstellen und Navigationsstrukturen sowie kurze Lebenszyklen auszeichnen (Gaedke, 1999).

Die Entwicklung derartiger Informationssysteme wird seit einigen Jahren technologisch durch eine Vielzahl unterschiedlicher Systeme wie z.B. Web-Editoren oder insbesondere Content-Management-Systeme unterstützt. Der Fokus dieser Technologien liegt im Wesentlichen auf der Gestaltung und flexiblen Erzeugung der grafischen Präsentation von Informationen, dem Einstellen und Verwalten von Inhalten z. T.

innerhalb eines Redaktionsprozesses sowie der Anbindung an Datenbanken und existierende Applikationen. Inzwischen sind zahlreiche webbasierte Informationssysteme entstanden, die Inhalte von Organisationen, Unternehmen, einzelnen Personen etc. im World Wide Web zur Verfügung stellen. Die Intention dieser Systeme liegt vorrangig in der Bereitstellung einer für den Besucher der Website attraktiven und bedienbaren Informationsplattform, die eine professionelle Außendarstellung erkennen lässt. Diese Zielsetzung wird jedoch häufig durch die zumeist hohe Komplexität der darzustellenden Infor-

mationen und einer Vielzahl von Navigationsmöglichkeiten, die eine Orientierung des Besuchers erschweren (Costagliola et al., 2002), verfehlt. Erschwerend kommt hinzu, dass im Unterschied zu konventionellen Applikationen webbasierte Anwendungen oft einer ständigen Überarbeitung sowohl hinsichtlich ihrer Struktur als auch ihrer Inhalte unterworfen sind. Eine Erneuerung des Informationsangebotes, z.B. aufgrund der Einbindung externer Dienste oder der Änderung der Navigationsstruktur, erfordert daher nicht selten eine weitgehende Neugestaltung der Anwendung.

Vor allem der Einsatz von Softwaretools wie HTML-Editoren, Database Publishing Wizards, Web Site Managers etc., die lediglich den Implementierungsaspekt berücksichtigen (Retschitzegger & Schwinger, 2000), begünstigte in den vergangenen Jahren eher eine Ad-hoc-Vorgehensweise als einen systematischen und methodischen Entwicklungsansatz. Selbst bei Web Content Management-Systemen konzentriert sich die methodische und technologische Unterstützung ausschließlich auf Teilaspekte des Lebenszyklus von Web-Applikationen, während die Verwendung von Entwurfsmodellen nicht Gegenstand derartiger Systeme ist. In der Folge sind webbasierte Anwendungen nicht nur in ihrer Entwicklung u. a. aufgrund fehlender Konzepte zur Wiederverwendbarkeit aufwändig, sondern vielfach auch kostenintensiv in Bezug auf Wartbarkeit und Pflege. Zudem genügen sie oftmals nur unzureichend den gesteckten Zielen der Betreiber und qualitativen Ansprüchen der Besucher.

Das Fehlen eines Engineering-Ansatzes bei der Entwicklung webbasierter Anwendungen ist insofern beachtlich, als dass starke Bezüge zum Software-Engineering existieren, die vor allem dadurch zu begründen sind, dass bei der Erstellung von Web-Anwendungen ebenfalls Programmierung und Software-Entwicklung von Bedeutung sind. Auch wenn Web-Anwendungen gegenüber konventionellen Applikationen eine Reihe von Eigenschaften aufweisen, die neue Anforderungen an die Entwurfs- und Realisierungsmethodik stellen, ist eine Übertragung der methodischen und werkzeuggestützten Vorgehensweisen bei der Softwareentwick-

lung auf die Web-Anwendungsentwicklung viel versprechend. Die Anwendung von Verfahren und Werkzeugen des Software-Engineering, insbesondere zur Unterteilung des Entwicklungsprozesses in aufeinander folgende Bearbeitungsschritte, hätte jedoch nicht nur positive Auswirkungen auf die Komplexität der Anwendungserstellung, sondern gleichermaßen auf die Qualität und die Wartbarkeit einer Web-Applikation.

Nicht zuletzt aus diesen Gründen sind vornehmlich aus dem akademischen Umfeld heraus inzwischen etwas mehr als zwanzig Modelle und Methoden zur Hypermedia- bzw. Web-Anwendungsentwicklung entstanden, die in mehreren Studien diskutiert werden (Christodoulou et al., 1998; Ceri et al., 1999; Fraternali, 1999; Koch, 1999; Costagliola et al., 2002; Woukeu, 2003). Dennoch zeigt sich, dass eine Übertragung der entwickelten Konzepte und Methoden auf die Entwicklungspraxis in Unternehmen bislang nicht stattgefunden hat und, falls überhaupt eine systematische Vorgehensweise zugrunde liegt, eigene Methoden favorisiert werden (Barry & Lang, 2001). Die geringe Akzeptanz vorhandener Methodologien trotz nach wie vor hoher Unterstützungsbedarfe seitens der Unternehmen zeigt, dass existierende Modellierungstechniken und Entwicklungsmethoden den gestellten Anforderungen nicht gerecht werden und für die betriebliche Praxis im Allgemeinen ungeeignet sind.

Von diesen Überlegungen ausgehend soll in diesem Beitrag eine Vorgehensweise für den komponentenorientierten Entwurf webbasierter Anwendungen vorgestellt werden, die auf Grundlage der Analyse und Anforderungsspezifikation webbasierter Anwendungen alle Modellierungsphasen in einem integrierten Ansatz unterstützt. Ausgehend von Entwurfsaspekten zur Informations- und Metadatenstruktur einer Anwendung werden in einem einheitlichen Ansatz Modellierungstechniken für den Entwurf eines Navigations-, Sichten- und Präsentationsmodells entwickelt. Die Entwurfsmethodik sieht die Modellierung der Anwendung auf der Grundlage einer grafischen Notation der verschiedenen Modellkomponenten vor, deren formale Beschreibung eine Überführung der

Entwurfsmodelle in einen Prototyp der Website ermöglicht, die im Sinne einer automatisierbaren Weiterverarbeitung erfolgt.

Neben der Unterstützung des Entwicklungsprozesses und der Ausrichtung auf eine weitestgehend automatisierte Weiterverarbeitung der vom Benutzer erstellten Modelle existieren weitere Anforderungen, die bei der Konzeption der Entwicklungsmethodik berücksichtigt werden müssen:

- Der Methodenverbund, der auf einen schrittweisen Aufbau der Web-Applikation ausgerichtet ist, muss Möglichkeiten zur frühzeitigen Evaluation zur Verfügung stellen, mit denen die Qualität der Navigationsstruktur und der Präsentation überprüft werden kann.
- Eine formale Beschreibung der verschiedenen Entwurfsmodelle muss die Überführbarkeit der vom Benutzer erstellten Modelle in den jeweils nachfolgenden Bearbeitungsschritt sicherstellen.
- Die verwendete grafische Beschreibung der Modelle sollte eine arbeitsfähige Notation darstellen, die für die unterschiedlichen Entwicklergruppen der verschiedenen Fachdisziplinen verständlich ist.
- Änderungen an Inhalten, Navigationspfaden bzw. der Darstellung sollten sowohl zur Modellierungszeit als auch zur Laufzeit unabhängig voneinander durchführbar sein.
- Die Modellierung sollte die Wiederverwendbarkeit von Teilen des Entwurfs ermöglichen.
- Der Methodenverbund muss hinsichtlich der enthaltenen Modellierungskomponenten vollständig sein, um alle Aspekte der Web-Anwendungsentwicklung abzudecken. Dies schließt insbesondere die Berücksichtigung extern verfügbarer Dienste und Inhalte, die an die Website angebunden werden sollen, und die Betrachtung von Kontextfaktoren ein.

Ein wesentliches Ziel ist es, die den Entwicklungsphasen zugehörigen Entwurfsmodelle derart zu gestalten, dass sie durch geeignete softwaretechnische Werkzeuge erstellt werden können und dadurch den Benutzer bei der Modellierung unterstützen.

2. Engineering-Ansatz für Web-Applikationen

Ausgangspunkt der hier vorgestellten Vorgehensweise ist eine Unterteilung des Entwicklungsprozesses webbasierter Anwendungen, der in Analogie zum Software-Engineering als *Web-Engineering* (Gellerson et al., 1997) bezeichnet wird: Er wird in die Phasen Analyse, Spezifikation, Konzeption und Betrieb und Wartung gegliedert. Neben einer weitestgehend übereinstimmenden Phaseneinteilung existieren jedoch Unterschiede sowohl zwischen Web-Applikationen und herkömmlichen Anwendungen als auch in den zugehörigen Entwicklungsprozessen:

- Web-Anwendungen sind überwiegend contentintensiv und enthalten oftmals Mischformen informationsvermittelnder und applikationsorientierter Teile mit vermehrt hoher multimedialer Ausprägung.
- Web-Applikationen unterliegen einer ständigen Veränderung bzw. Erweiterung der dargestellten Inhalte und Funktionalitäten.
- Visuelle Eigenschaften nehmen bei der Entwicklung von Web-Anwendungen einen hohen Stellenwert ein. Neben der Programmierung sind auch kreative bzw. künstlerische Aspekte bzgl. des Designs von Bedeutung.
- Web-Anwendungen berücksichtigen verstärkt unterschiedliche Besucherrollen, die den Benutzern eine an ihre Interessen und Aufgaben ausgerichtete inhaltliche Darstellung ermöglichen. Benutzerprofile erlauben darüber hinaus die Einbeziehung individueller Interessen.
- Web-Applikationen unterliegen oftmals kurzen Entwicklungszeiten, so dass vor allem sowohl der Analyse- und Konzeptionsphase als auch der Testphase nicht die erforderliche Bearbeitungszeit beigemessen wird.
- Gewöhnlich sind an Web-Projekten Mitarbeiter verschiedener Disziplinen beteiligt, die unterschiedliche Ausrichtungen, Erfahrungen und Kenntnisse haben.

Bild 1 zeigt ein Modell für den Entwicklungsprozess von Web-Anwendungen, das alle relevanten Entwurfsbereiche und Prozessschritte abdeckt. Aus-

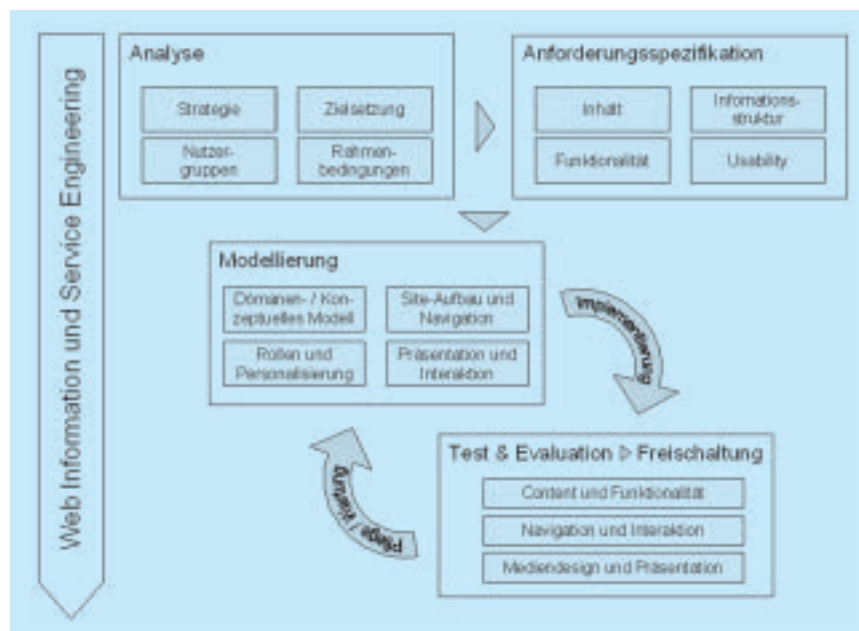


Bild 1: Phasen im Web Engineering

gangspunkt ist eine Analyse grundsätzlicher Faktoren. Hierzu zählen die Definition der Ziele, die mit der Website erreicht werden sollen, die Identifikation relevanter Nutzergruppen, die Untersuchung der gegebenen Rahmenbedingungen und die Festlegung der strategischen Ausrichtung. In einem weiteren Schritt werden Inhalt und Funktionalität der Website spezifiziert und erste Richtlinien für die Strukturierung der Informationen sowie Bestimmungen zur Ergonomie festgelegt. Darauf aufbauend erfolgt der Entwurf verschiedener Modelle, die sich neben der Kategorisierung des Informationsbestands sowohl mit dem strukturellen Aufbau und dem Design der Website als auch mit der Definition der unterschiedlichen Benutzerrollen befassen. Die Modelle stellen den Ausgangspunkt für das Prototyping bzw. die Implementierung der Web-Anwendung dar und werden in einem iterativen Prozess in Bezug auf qualitative und ergonomische Merkmale verifiziert und ggf. optimiert.

Innerhalb der Modellierungsphase dient die *konzeptuelle* Ebene der Erstellung einer Themenstruktur, die wesentliche Konzepte und Beziehungen zwischen diesen enthält. Die Themenstruktur beschreibt den Informationsraum der späteren Website und dient als Grundlage für den systematischen Entwurf einer Navigations- und Sichten-

struktur. Darauf aufbauend beinhaltet die *navigationale* Ebene eine Beschreibung der darzustellenden Inhalte sowie der Navigationsstrukturen, die den Zugang zu den Informationen definieren. Das daraus resultierende Navigationsmodell wird innerhalb der *Sichtenebene* um die Definition der tatsächlich darzustellenden Inhalte in Abhängigkeit verschiedener Nebenbedingungen, wie beispielsweise die Rolle des Benutzers, seine Interessen, Ortsinformationen etc., erweitert. Zuletzt wird die grafische Präsentation der Inhalte in der *Präsentationsebene* festgelegt.

Aus der Betrachtung unterschiedlicher Ebenen in Verbindung mit der Generierung verschiedener Modelle am Ende jeder Phase ergeben sich Vorteile für den Entwickler bzgl. Aufbau und Wartung der Web-Anwendung: Im Hinblick auf Möglichkeiten zur Unterstützung der Kooperation und Koordination der an der Entwicklung beteiligten Akteure lassen sich frühzeitig Teilergebnisse kommunizieren. Darüber hinaus bestehen Verifikationsmöglichkeiten am Ende jeder Phase. Aufgrund der Trennung von Inhalt, Navigation und Darstellung ist es zudem möglich, die zugrunde liegenden Daten unabhängig von Navigationspfaden oder Darstellungsmerkmalen zu warten, d.h. ohne die Web-Anwendung neu implementieren zu müssen. Inhalt, Navigation und Präsen-

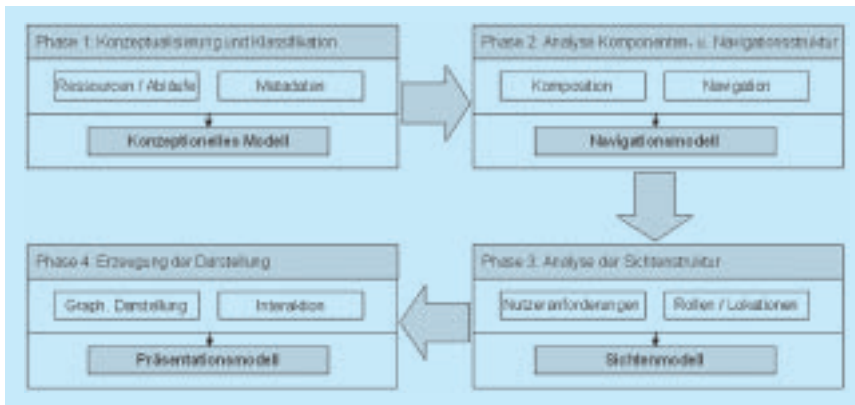


Bild 2: Entwurfsbereiche innerhalb der Modellierungsphase

tation können sowohl zur Laufzeit (z. B. für Fein-Tuning) als auch beim Redesign unabhängig voneinander modifiziert werden.

Die Modellierungsphase einer Web-Anwendung setzt sich demnach aus folgenden Schritten zusammen: Entwurf des Metadatenmodells (Themenstruktur), Aufbau des Navigationsmodells, formale Analyse der Benutzeranforderungen, Ableiten der Sichtenstruktur zur Modellierung personalisierter bzw. rollenbasierter Darstellungen auf der Grundlage des Navigationsmodells, Konfiguration externer Dienste und Definition der grafischen Darstellung und Interaktion (vgl. Bild 2). Wichtig ist, dass die jeweiligen Metamodelle, die sich auf die verschiedenen Entwurfsbereiche einer Web-Anwendung beziehen, aufgrund ihrer Konzeption eine weitestgehend automatisierte Weiterverarbeitung ermöglichen, die eine systemunterstützte Prototypengenerierung der Web-Applikation einschließt. Dem Entwickler stehen hierdurch Methoden zur Verfügung, mit denen er die von ihm entworfenen Modelle anhand des erstellten Prototyps in Bezug auf Inhalt und Funktionalität prüfen sowie die Effizienz der Struktur, Navigation und Präsentation evaluieren kann. Die entsprechenden Untersuchungen sollen sich dabei auf einzelne Modelle beschränken können, so dass sich im Sinne einer iterativen Vorgehensweise Teilbereiche des Modellierungsprozesses evaluieren und optimieren lassen.

Betrachten wir die Entwurfsmodelle der verschiedenen Ebenen nun im Einzelnen.

2.1 Konzeptionelles Modell

Die Ressourcen, die als Instanzen in Form von Dokumenten, Aufgaben- und Prozessbeschreibungen etc. vorliegen, dienen als Grundlage für die Erstellung des konzeptionellen Modells. Dieses setzt sich aus einer übergeordneten Themenstruktur, bestehend aus Konzepten und zugehörigen Assoziationen, sowie der Zuordnung von Instanzen aus der Ressourcenmenge zusammen. Das konzeptionelle Modell stellt die Konzeptualisierung des Informationsbestandes dar und ermöglicht in Form einer Ontologie die Zugriffsmöglichkeit auf die instanziierten Inhalte. Die Herleitung der Themenstruktur kann prinzipiell auf zwei unterschiedliche Arten erfolgen. Zum einen können solche Begriffssysteme manuell erarbeitet und in einem weiteren Schritt den darin enthaltenen Themen die instanziierten Inhalte zugewiesen werden. Zum anderen besteht die Möglichkeit, die vorhandenen Dokumente automatisch zu klassifizieren, wobei allerdings eine manuelle Nachbearbeitung unerlässlich ist. Das konzeptionelle Modell stellt eine Navigationsmöglichkeit über Themen zur Verfügung, d. h. über die Assoziationen zwischen den Themen ist es möglich, innerhalb der Ontologie zu navigieren bzw. diese zu explorieren. Diese Form der Navigation unterscheidet sich jedoch gegenüber der in einer Web-Anwendung definierten Navigation, deren Navigationsrelationen bzw. Links nicht den Assoziationen innerhalb der Ontologie entsprechen müssen. Allerdings lassen sich mithilfe einer Ontologie explorative Zugriffe innerhalb der Web-Applikation realisieren und darüber hinaus Navigationsmöglichkeiten

auf dynamische Mengen und Hierarchien erweitern.

2.2 Komponentenentwurf und -strukturierung

Der Aufbau eines Navigationsmodells für eine Website erfolgt nach dem hier vorgestellten Ansatz komponentenorientiert. Dabei werden zwei unterschiedliche Arten von Komponenten eingesetzt: Strukturelle Komponenten bestimmen den strukturellen Aufbau einer Seite, d. h. sie definieren die Zusammensetzung von Komponenten zu Teilbereichen und deren konzeptionelle Anordnung innerhalb der Seite. Inhaltsspezifische Komponenten sind für die Präsentation der Inhalte verantwortlich. Diese Komponenten legen die Darstellung statischer Inhalte, d. h. Inhalte, die bereits zur Modellierungszeit bekannt sind, sowie dynamischer Inhalte, die sich erst zur Laufzeit der Web-Anwendung ergeben, fest. Die zum Aufbau von Navigationsmodellen wesentlichen Komponenten sollen im Folgenden kurz vorgestellt werden.

2.3 Container und Partitionen

Die Elemente, die eine einzelne Website beschreiben, werden in sog. *Containern* organisiert. Diese dienen in erster Linie dazu, sowohl die gesamte Website als auch einzelne Websites zu strukturieren und die darzustellenden Inhalte getrennt voneinander zu betrachten. Durch Einbettung weiterer Container in einen Container lassen sich verschachtelte Websites darstellen. Jeder Container kann beliebig viele Subcontainer beinhalten, die wiederum als eigenständige Container betrachtet werden, und definiert einen Zielbereich, in dem die Inhalte der Web-Anwendung mittels einer Modellklasse angezeigt werden können.

Durch die Möglichkeit, Container ineinander einzubetten, entstehen Ebenen. Mehrere Subcontainer innerhalb eines Containers, die auf der gleichen Ebene liegen, werden gemäß einer XOR-Verknüpfung dargestellt, d. h. in diesem Fall ist immer genau ein Subcontainer sichtbar (vgl. Bild 3 (a)). Liegen mehrere Subcontainer auf gleicher Ebene eines Containers, so muss festgelegt werden, welcher Subcontainer zunächst sichtbar ist. Dazu wird gemäß Bild 4 (a) ein Startzustand definiert. Um verschiedene Container parallel darstellen zu können,

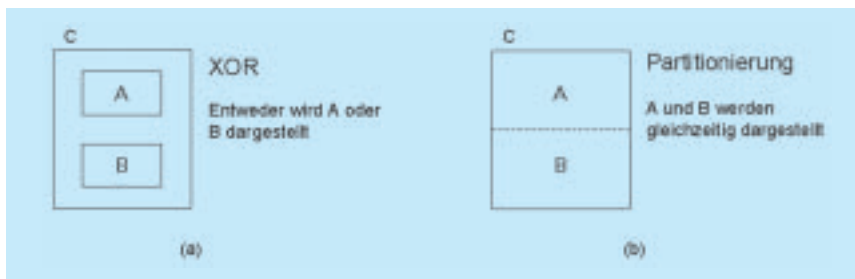


Bild 3: Container und Partitionen

werden Subcontainer als einzelne Partitionen angeordnet. Alle Partitionen in einem Container werden so in der Web-Anwendung gleichzeitig angezeigt (vgl. Bild 3 (b)).

Die Inhalte, die in einem Container beschrieben werden können und später in der Web-Darstellung präsentiert werden, lassen sich in zwei Kategorien einteilen. Zum einen stehen sie zur Modellierungszeit bereits fest, es handelt sich in diesem Fall um statische Inhalte. Zum anderen ist es möglich, dass sich die auf einer Website anzuzeigenden Informationen erst zur Laufzeit einer Web-Anwendung ergeben, also dynamisch erzeugt werden. Sowohl statische als auch dynamische Inhalte werden nicht unmittelbar in einem Container gehalten, sondern durch sog. Modellklassen beschrieben, die einem Container zugeordnet sind. Prinzipiell kann jeder Container neben Subcontainern beliebig viele Modellklassen enthalten.

2.4 Platzhalter

Mithilfe von Platzhaltern können Teilbereiche der Web-Anwendung zur besseren Überschaubarkeit schematisch dargestellt werden (vgl. Bild 4 (b)). Eine detaillierte Modellierung des durch einen Platzhalter nur stellvertretend dargestellten Bereiches kann an einer anderen Stelle erfolgen.

2.5 Primitive Klassen

Primitive Klassen beschreiben einfache Objekte, die nicht weiter zerlegt werden können. Hierzu zählen beispielsweise Texte, Bilder, Multimediaobjekte, externe Links, einfache HTML-Seiten etc. Derartige Objekte werden als Instanzen ihres zugehörigen primitiven statischen Konzeptes innerhalb der zugrunde liegenden Ontologie betrachtet und dienen in erster Linie zur einheitlichen Beschreibung der verschiedenen darzustellenden Ob-

jekte innerhalb des Navigationsmodells. Da nicht alle darzustellenden Inhalte bereits zur Modellierungszeit feststehen, wird zwischen primitiven statischen Klassen und primitiven dynamischen Klassen unterschieden. Bei letzteren handelt es sich um Modellkomponenten, die ein Attribut einer Instanz eines Konzeptes der Ontologie darstellen können, die erst zur Laufzeit definiert wird.

2.6 Navigationale Klassen

In einem Zielbereich kann entweder eine einzelne Instanz eines Konzeptes dargestellt werden oder, falls zu einem Konzept mehrere Instanzen vorhanden sind, eine Übersicht der vorhandenen Instanzen mit der Möglichkeit, diese an definierter Stelle im Einzelnen zu betrachten. Zu diesem Zweck werden navigationale Klassen eingeführt, die die Navigation auf einer Menge von Instanzen mithilfe einer Übersicht ermöglichen. Die Art und Weise, in der die Übersicht dargestellt wird, ist im zugrunde liegenden Navigationsmodell bereits festgelegt und wird hierzu aus einer Menge vordefinierter Navigationstypen gewählt. Zu dieser Menge gehören Darstellungstypen wie beispielsweise Baumstruktur, Listenstruktur, Sequenz etc. (vgl. Bild 5).

Die grafische Darstellung einer navigationalen Klasse besteht aus zwei Kompo-

nenten: der navigationalen Klasse, die einem Container zugeordnet ist, und einem Zielcontainer. Beide Komponenten sind in Abhängigkeit von der Tiefe des Suchfeldes durch eine oder mehrere Navigationsrelationen miteinander verbunden. Bei den Navigationsrelationen handelt es sich um parametrisierbare Links, die durch die einzelnen Instanzen der navigationalen Klasse beim entsprechenden Aufruf initialisiert werden. Bild 6 zeigt die grafische Darstellung einer navigationalen Klasse einschließlich der Parameter, die zur Modellierungszeit angegeben werden müssen. Die Navigationsrelationen sind in der grafischen Darstellung durch eine offene Pfeilspitze gekennzeichnet.

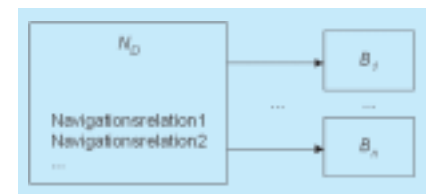


Bild 6: Grafische Darstellung einer navigationalen dynamischen Klasse

2.7 Navigationsmodellierung

Zur Modellierung der Navigationsstruktur werden zunächst sog. Kompositionsmodelle erstellt, die die Zusammensetzung beliebiger Komponenten zu einzelnen Websites definieren. Das Navigationsmodell setzt sich darauf aufbauend aus den Kompositionsmodellen der einzelnen Websites und einer zusätzlichen übergeordneten Navigationsstruktur zusammen. Diese legt die Navigation zwischen den einzelnen Komponenten und Websites fest.

Die grafische Beschreibung des Navigationsmodells verwendet einen auf der



Bild 4: (a) Festlegung des Startzustandes; (b) Platzhalter für einen an anderer Stelle definierten Container



Bild 5: Auswahl unterschiedlicher Typen navigationaler Klassen

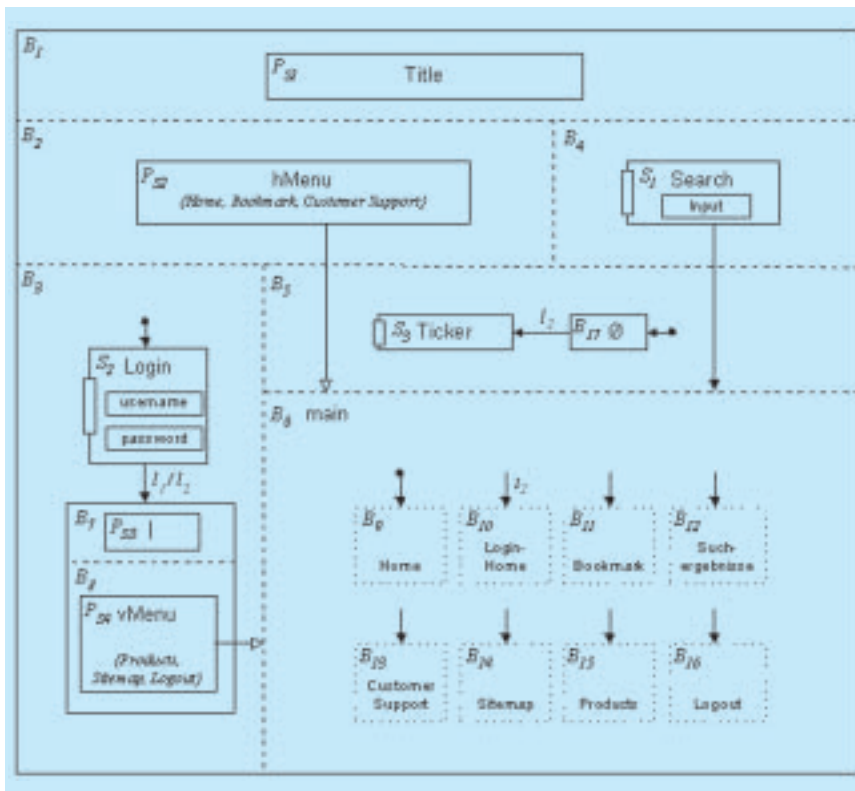


Bild 7: Navigationsmodell (Auszug aus einem Portal)

Statechart-Notation (Harel, 1987) basierenden Ansatz. Hierbei werden atomare und nicht-atomare Komponenten sowie Container und Partitionen durch gerichtete Navigationsrelationen miteinander verbunden. Der Bereich, der später im Webbrowser für die Darstellung einer

oder mehrerer Komponenten zur Verfügung gestellt wird, lässt sich mithilfe von Containern und Partitionen festlegen. Hierbei können sowohl konjunktive als auch disjunktive Sichtenstrukturen beschrieben werden. Konjunktive Sichtenstrukturen, bei denen alle unmittelbar

enthaltenen Komponenten gleichzeitig sichtbar sind, entstehen durch Partitionierung eines Containers. Hingegen bilden mehrere Komponenten gleicher Ebene in einem unpartitionierten Bereich eine disjunktive Sichtenstruktur.

Bild 7 zeigt beispielhaft das Navigationsmodell zu einer Portalseite eines Unternehmens, das den Rahmen der entsprechenden Anwendung darstellt. In der gesamten Web-Applikation findet sich im oberen Bereich eine Titelzeile, unmittelbar darunter liegt eine durch eine Linkkollektion realisierte Menüleiste mit einem rechts angehefteten Suchfeld in Form einer Applikation. Unterhalb dieser beiden Komponenten befindet sich eine weitere Partition. Diese enthält im Startzustand links zunächst ein Login-Feld, das nach erfolgreicher Anmeldung durch ein vertikales Menü mit einem darüber liegenden Bild („I“ steht hier für „Image“) ersetzt wird. Rechts davon wird im oberen Bereich zunächst nichts angezeigt („Ø“ steht hier für die leere Menge). Erst nachdem die Anmeldung abgeschlossen ist, wird der Ticker aktiviert. Der untere Bereich stellt den Hauptbereich der Anwendung dar. Er ist gleichzeitig der Zielbereich für die beiden Menüleisten und das Suchfeld. Die verschiedenen Komponenten, die in diesem Bereich dargestellt werden, können nun an anderer Stelle genauer spezifiziert werden.

In einem weiteren Beispiel soll nun die Modellierung dynamischer Inhalte, die als Instanzen in der zugrunde liegenden Ontologie vorliegen, verdeutlicht werden. Hierzu werden sowohl primitive dynamische Klassen als auch navigationale Klassen eingesetzt. Bild 8 zeigt die grafische Beschreibung des Kompositionsmodells für die Komponente „Products“ aus Bild 7.

Ausgehend von der Homepage des Unternehmens erhält der Benutzer im linken Bereich eine Übersicht über die Produkte in Form der Produktnamen. Im rechten Bereich wird zu einem im Produktnavigator ausgewählten Produkt zunächst die Produktbeschreibung angezeigt. Von hier aus können über die Navigationsrelation stellvertretend auch Testergebnisse zusammen mit einer Auflistung von Anwendungen präsentiert werden. Für eine ausführlichere Dokumentation einer im Anwendungsnavigator gewählten

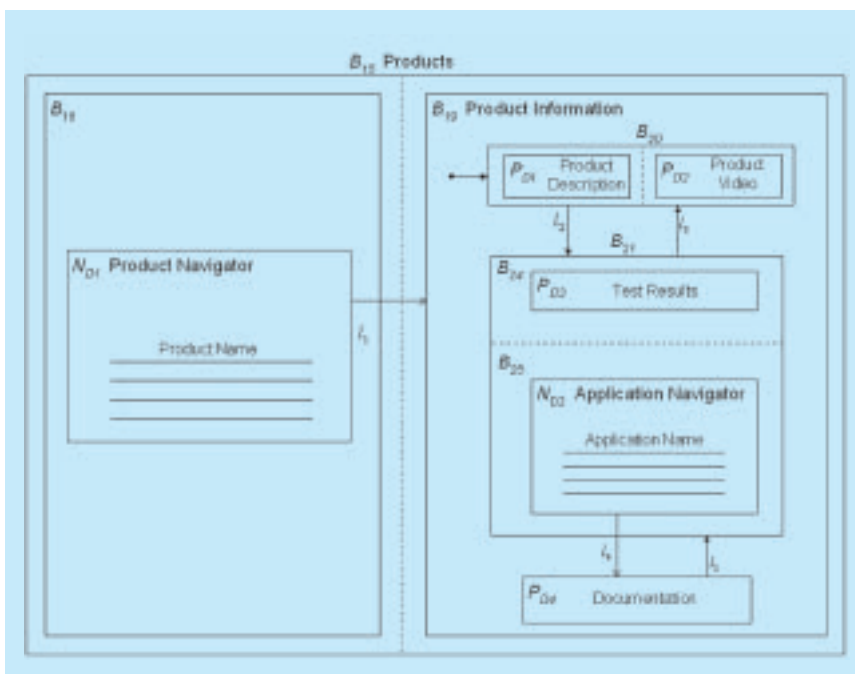


Bild 8: Grafische Beschreibung der Komponente „Products“

Anwendung steht die Navigationsrelation zur Verfügung, die dem Benutzer die gewünschte Anwendungsdokumentation liefert.

2.8 Sichten- und Kontextmodellierung

Um die verschiedenen Möglichkeiten zur Personalisierung beim Entwurf einer Web-Applikation methodisch zu unterstützen, bedarf es der Modellierung von Zugriffs- und Sichtenstrukturen. Das Navigationsmodell liefert eine grafische und formale Beschreibung der Struktur und der thematischen Inhalte einer Website und repräsentiert eine einheitliche Sicht auf die darzustellenden Informationen, unabhängig vom Kontext, in dem der Benutzer auf die Inhalte zugreift. Die zusätzliche Berücksichtigung von Kontextfaktoren wie beispielsweise die Rolle des Benutzers, sein gegenwärtiger Aufenthaltsort, der Zeitpunkt und das von ihm zur Interaktion genutzte Gerät ist jedoch notwendig, um ein den Anforderungen des Benutzers angepasstes Informationsangebot zu erzeugen.

Die Kontextmodellierung lässt sich prinzipiell in drei Bereiche gliedern (vgl. Schmidt et al., 1999): Der technische Kontext beinhaltet technische Rahmenbedingungen, wie Hard- und Softwareeigenschaften des benutzten Endgerätes, Eigenschaften des Netzwerks und den Zustand der Anwendung selbst. Diese Angaben sind notwendig, um bei der Bereitstellung von Inhalten durch die Web-Anwendung Faktoren wie z. B. Displaygröße und Übertragungsrate zu berücksichtigen. Orts- und zeitbezogene Informationen sind im natürlichen Kontext enthalten. Hierdurch erhält die Web-Anwendung einerseits die Möglichkeit, Informationen in Abhängigkeit des gegenwärtigen Standortes zu sortieren, und andererseits Gültigkeitszeiträume in die Auswahl der darzustellenden Inhalte einzubeziehen. Der soziale Kontext umfasst schließlich die Beschreibung des Benutzers bzw. der Rollen, die der Benutzer einnimmt.

Jeder Modellklasse des Navigationsmodells kann eine *Zugriffsklasse* zugeordnet werden. Die Modellklassen definieren lediglich die Inhalte, die in den entsprechenden Containern dargestellt werden sollen, jedoch keine Bedingungen für den Zugriff auf diese Inhalte.

Dies bedeutet, dass Nebenbedingungen, die sich aus der Berücksichtigung des Kontextes ergeben, im Navigationsmodell nicht aufgenommen werden. Die Aufgabe der Zugriffsklassen liegt nun darin, diese Nebenbedingungen festzulegen und dementsprechend den Zugriff auf die Inhalte und somit ihre Sichtbarkeit in Abhängigkeit von unterschiedlichen Benutzerrollen zu definieren. Zur Betrachtung verschiedener Kontexte in der Web-Anwendung werden ggf. mehrere Zugriffsklassen für eine Modellklasse benötigt.

Eine Zugriffsklasse setzt sich aus zwei Teilen zusammen: einer Zuordnungsangabe und einem Bedingungsteil. Die Zuordnungsangabe legt den Benutzer bzw. die Rolle fest, auf den sich die Zugriffsklasse bezieht. Der Bedingungsteil enthält einen beliebigen booleschen Ausdruck, der erfüllt sein muss, um den in der zugehörigen Modellklasse festgelegten Inhalt darzustellen. Bild 9 zeigt beispielhaft die Verwendung einer Zugriffsklasse zur Darstellung aller Restaurants, die sich in unmittelbarer Umgebung des Benutzers befinden und gleichzeitig noch geöffnet sind.

2.9 Präsentationsmodellierung

Das Präsentationsmodell dient der Transformation des konzeptuellen Schemas auf eine Lower-Level-Präsentation und enthält im Wesentlichen die zur grafischen Darstellung notwendigen Informationen. Hierzu zählen genaue Angaben über die Position und die grafische Gestaltung der darzustellenden Inhalte und Interaktionsflächen. Diese Informationen werden unter Verwendung von Stylesheets den einzelnen Sichtenklassen zugeordnet und zur Laufzeit abge-

rufen. Neben diesen Informationen zur grafischen Gestaltung der einzelnen Komponenten können weitere Angaben im Präsentationsmodell festgelegt werden. Das Präsentationsmodell stellt hierzu eine offene Struktur zur Verfügung, die es erlaubt, beliebige Merkmale der graphischen Gestaltung zu bestimmen. Beispielsweise lassen sich zusätzlich Angaben über die zu verwendende Schriftart, die Schriftgröße etc. festlegen.

Die grafische Modellierung einer Präsentationsklasse erfolgt analog zur Verwendung von Zugriffsklassen. Präsentationsklassen beziehen sich ebenfalls auf bestimmte Benutzergruppen bzw. einzelne Benutzer. Prinzipiell können jeder Modellklasse beliebig viele Präsentationsklassen zugeordnet werden. Wird eine Präsentationsklasse statt einer Modellklasse einem Container zugewiesen, so beziehen sich die Darstellungseigenschaften auf die unmittelbar im entsprechenden Container enthaltenen Modellklassen. Navigationsrelationen mit einem Container als Ursprung können in Analogie zur Verwendung von auch Präsentationsklassen zugeordnet werden.

Bild 10 zeigt beispielhaft die grafische Modellierung zweier Präsentationsklassen, die einer Modellklasse bzw. einem Container zugeordnet wurden. Die grafischen Merkmale sowohl des Navigators auf der linken Seite als auch des Informationsbereiches auf der rechten Seite gelten für alle Benutzer. Unterschiede zwischen den beiden Partitionen bestehen in der Größendefinition und der Farbwahl. Während in diesem Beispiel dem Navigationsbereich dreißig Prozent des horizontalen Bereichs der Komponente zugewiesen werden, bleiben dem Informationsbereich die übr-

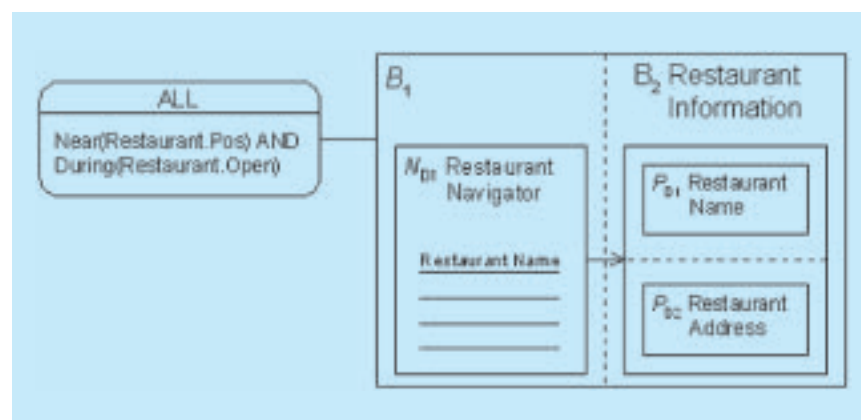


Bild 9: Zugriffsklasse zur orts- und zeitabhängigen Präsentation

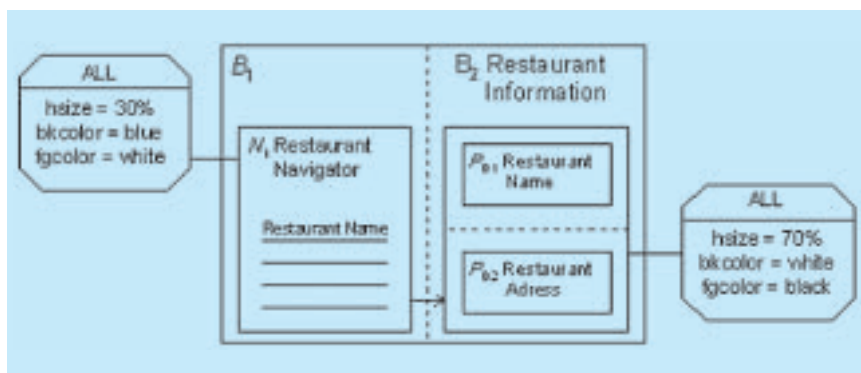


Bild 10: Verwendung von Präsentationsklassen

gen siebzig Prozent zur Darstellung des Namens und der Adresse eines gewählten Restaurants.

3. Erfahrungen und Ausblick

Bislang existiert noch kein marktfähiges softwarebasiertes Modellierungswerkzeug, das die hier vorgestellte Modellierungsmethodik unterstützt, sondern lediglich eine prototypische Umsetzung, die die prinzipielle Funktionsweise demonstriert (Häusser et al., 2003). Die bisher durchgeführten Untersuchungen beziehen sich daher weniger auf den gesamten Entwicklungsprozess einschließlich Implementierung und Pflege als auf die Verständlichkeit und Anwendbarkeit der verschiedenen Modelle. Ziel dieser Untersuchungen war es primär, das Vorgehen beim Entwurf von Websites mit dem Modellierungsansatz des Methodenverbundes zu vergleichen.

Die Aufgabenstellung für eine Gruppe von Testpersonen umfasste zunächst den Entwurf einer schematischen Darstellung zu einer konkreten Website, der schrittweise ergänzt wurde. Gegenstand der anschließenden Befragung war die Anwendung der einzelnen Entwurfsmodelle des Methodenverbundes. Auch wenn sowohl die Effizienz als auch die Konsistenz der verschiedenen Lösungsansätze unterschiedlich zu bewerten ist, zeigten alle Modelle eine starke Orientierung an der tatsächlichen Darstellung einer Website. Der überwiegende Teil der Probanden favorisierte einen Ansatz, der hinsichtlich disjunktiver Teilbereiche von der hohen Ausrichtung auf die Präsentation der Seite abstrahierte. Hier zeigten sich Unterschiede in den einzelnen Modellen in erster Linie bei der gra-

fischen Notation der disjunktiven Darstellung.

Bei der Erstellung der schematischen Darstellungen bestand die übliche Vorgehensweise seitens der Testpersonen darin, konjunktive Bereiche nicht zusätzlich, z. B. mittels einer gestrichelten Linie, voneinander zu trennen. Die Bereiche wurden stattdessen als umrandete Komponenten modelliert oder mithilfe einer durchgezogenen Linie unterteilt. Auch wenn diese Methodik weitgehend einem intuitiven Vorgehen entspricht, beurteilte der überwiegende Teil bei einer direkten Gegenüberstellung die auf Grundlage der hier vorgestellten Notation (WISE-Notation) erstellten Modelle im ersten Eindruck positiver. Die Bewertung bezog sich dabei auf die Kriterien Strukturiertheit, Verständlichkeit und Ähnlichkeit zu einer Website.

Ohne eine genaue Kenntnis über die WISE-Notation zu besitzen, wirkten die unter Verwendung dieser Notation erstellten Modelle im Vergleich zum intuitiven Ansatz der Testpersonen nur geringfügig strukturierter, jedoch auch etwas weniger verständlich. Die geringe Abweichung lässt darauf schließen, dass die Entwurfsmodelle der WISE-Notation den Benutzer auf den ersten Blick nicht überfordern. Unter Kenntnis beider Notationen wirkten sie sogar deutlich strukturierter und verständlicher gegenüber der intuitiven Vorgehensweise.

Dieser Eindruck wird durch die Ergebnisse der Anwendung der Entwurfsmodelle unterstrichen. Zu einem vorgegebenen Modell (s. Bild 8) sollten nach einer maximal vierminütigen Erläuterung die entsprechenden Websites hinsichtlich Aufbau, Struktur und Navigation skizziert werden. Die Aufgabe wurde von allen Probanden fehlerfrei gelöst. Dementsprechend empfanden fast alle

Probanden die WISE-Notation als eher unkompliziert und relativ leicht erlernbar.

Literatur

- Barry, C.; Lang, M.: A Survey of Multimedia and Web Development Techniques and Methodology Usage. *IEEE Mul-timedia* **8(3)** (2001) 52–60.
- Ceri, S.; Fraternali, P.; Paraboschi, S.: Design Principles for Data-Intensive Web Sites. *SIG-MOD Record* **24**, No. 1, March 1999.
- Costagliola, G.; Ferrucci, F.; Francese, R.: Web engineering: Models and methodologies for the design of hypermedia applications. In: *Handbook of Software Engineering & Knowledge Engineering*, volume 2. (Ed. Chang, S. K.), Emerging Technologies, pages 181–199. World Scientific, 2002.
- Christodoulou, S.; Styliaras, G.; Papatheodourou, T.: *Evaluation of Hypermedia Application Development and Management Systems*. Proc. of the 9th ACM Conference on Hypertext and Hypermedia. Pittsburgh, 1998.
- Fraternali, P.: *Tools and approaches for data-intensive Web applications: A survey*. ACM Computing Surveys Vol. **31**, No. 3, September 1999.
- Gaedke M.: *Wiederverwendung von Komponenten in Web-Anwendungen*. In: *Tagungsband des 1. Workshops Komponentenorientierte betriebliche Anwendungssysteme (WKBA 1)* (Hrsg. Turowski, K.) Universität Karlsruhe; 1999. Magdeburg, S. 31–36, 1999.
- Gellersen, H.; Gaedke, M.: An Object-Oriented Model for the Web Application Development Process. *IEEE Internet Computing*, January – Feb 1999.
- Harel, D.: Statecharts: A Visual Formalism for Complex Systems. In: *Science of Computer Programming*. Elsevier Science Publishers. North Holland, 1987. S. 231–274.
- Häusser, T.; Mirochnitchenko, M.; Rindermann, M.: Web Ontology Management Application. In: *Mensch & Computer 2003: Interaktion in Bewegung*. (Hrsg. Szwillus, G.; Ziegler, J.) Stuttgart: B. G. Teubner, 2003. S. 247–256.
- Koch, N.: *A Comparative Study of Methods for Hypermedia Development*. Technical Re-

port 9905, Ludwig-Maximilians-Universität München, November 1999.

Retschitzegger, W.; Schwinger, W.: *Towards Modeling of DataWeb Applications - A Requirements' Perspective*. Proc. of the Americas Conf on Information Systems (AMCIS), California, Vol. I, Aug. 2000.

Schmidt, A.; Beigl, M.; Gellerson, H. W.: There is more to Context than Location. *Computers & Graphics Journal* Vol. **23**, No. 6, December 1999.

Woukeu, A.; Carr, L.; Wills, G.; Hall, W.: *Rethinking Web Design Models: Requirements for Addressing the Content*. Technical Report ECSTR-IAM03 -002, Department of Electronics and Computer Science, University of Southampton, 2003.



Michael Wissen, wissenschaftlicher Mitarbeiter des Fraunhofer Instituts für Arbeitswirtschaft und Organisation (IAO) in Stuttgart, arbeitet in verschiedenen öffentlich sowie privatwirtschaftlich geförderten Projekten an Fragestellungen u. a. aus den Bereichen Web Application Engineering und Mensch-Technik-Interaktion.
E-Mail: michael.wissen@iao.fhg.de.