

WebComposition Process Model: Ein Vorgehensmodell zur Entwicklung und Evolution von Web-Anwendungen

Martin Gaedke, Guntram Gräf

Telecooperation Office (TecO), Universität Karlsruhe, Vincenz-Prißnitz Str. 1, 76131 Karlsruhe, Deutschland, Tel.: +49 (721) 6902-79, Fax: -16, E-Mail: {gaedke|graef}@teco.uni-karlsruhe.de, URL: <http://www.teco.uni-karlsruhe.de>

Zusammenfassung: Das World Wide Web ist aus Sicht der Softwaretechnik eine neue Anwendungsplattform. Das dem Web zugrunde liegende Implementierungsmodell erschwert die Anwendung klassischer Vorgehensmodelle für die Entwicklung und insbesondere Evolution von Web-Anwendungen. Die Vorteile komponentenbasierter Softwareentwicklung scheinen durch die im Web vorherrschende Innovationsdynamik besonders für die Entwicklung und Evolution von Web-Anwendungen geeignet zu sein, jedoch erfordert eine solche Vorgehensweise eine dedizierte Unterstützung. Das WebComposition Vorgehensmodell tritt dieser Forderung entgegen, indem es die komponentenbasierte Entwicklung von Web-Anwendungen beschreibt. Es nutzt hierzu eine XML-basierte Auszeichnungssprache, um den Anforderungen des Webs gerecht zu werden. Für die Koordination der Komponenten wird das Konzept des offenen Vorgehensmodells und der expliziten Wiederverwendungsunterstützung eingeführt. Der planbaren Evolution von Web-Anwendungen begegnet das Vorgehensmodell letztendlich durch die Beschreibung von Anwendungsdomänen mittels Fachkomponenten.

Schlüsselworte: Web Engineering, WebComposition, Komponenten, Wiederverwendung, Evolution

1 Einleitung

Durch den ubiquitären Zugriff auf Informationen und Anwendungen jeglicher Art konnte sich das World Wide Web (Web) in kürzester Zeit als *die* Plattform für die Auslieferung von Hypermedia-Anwendungen etablieren. Unter dem Einfluß des zunehmenden Wettbewerbs, insbesondere im kommerziellen Bereich, dem elektronischen Handel, unterliegen diese Anwendungen einem ständigen Wandel, sei es bezüglich Funktionalität, Anwendungsschnittstellen oder der verfügbaren Informationen (Cusumano/Yoffie 1999). Diese Anwendungen, im folgenden auch *Web-Anwendungen* genannt, sind in besonderem Maße durch Besonderheiten des ihnen zugrundeliegenden Implementierungsmodells geprägt. Bedingt durch die hohe Innovationsgeschwindigkeit werden die Lebenszyklen von Web-Anwendungen immer kürzer, da sich die Anwendungen ständig einem evolutionären Prozeß unterziehen müssen. Der zunehmenden Komplexität von Anwendungen im Web wird jedoch nur durch eine wenig disziplinierte Vorgehensweise für Entwicklung und Evolution begegnet (Barta/Schranz 1998; Gellersen/Gaedke 1999).

Die Erkenntnis, daß Erstellung und Evolution von Anwendungen im World Wide Web ähnliche Unterstützung benötigen, wie sie für herkömmliche Anwendungen durch die Modelle, Methoden und Prinzipien der Softwaretechnik gegeben werden, legt die Erarbeitung softwaretechnischer Grundlagen nahe, denn das World Wide Web mit seinem besonderen Charakter und Eigenschaften ist aus Sicht der Softwaretechnik eine neue Anwendungsdomäne (Gaedke 1998). Diese neue Disziplin, die sich in den letzten zwei Jahren als *Web Engineering*

etabliert hat, verspricht eine Aufwandsreduktion und Qualitätssteigerung bei Entwicklung und Evolution von Web-Anwendungen:

Web Engineering – Die Anwendung systematischer, disziplinierter und quantifizierbarer Ansätze für die kosteneffektive Entwicklung und Evolution von qualitativ hochwertigen Anwendungen im World Wide Web.

Das Web Engineering berücksichtigt dabei insbesondere implizit die zentrale Forderung von Berners-Lee nach *Heterogenität* des Systems und *autonomer Verwaltung* der Ressourcen (Berners-Lee 1990). Diese Forderung, die im folgenden als *Grundprinzipien des Webs* bezeichnet werden, stellt eine besondere Hürde heutiger Ansätze zur Entwicklung und Wartung von Web-Anwendungen dar, wie im zweiten Abschnitt deutlich wird. Web-Anwendungen erfordern darüber hinaus eine feingranulare und wiederverwendungsorientierte Implementierung, damit existierende Anwendungen oder Teile davon in neue Anwendungen integriert oder zu neuen Anwendungen föderiert werden können (Barta/Schranz 1998; Schwabe/Rossi 1998; Gellersen/Wicke/Gaedke 1997).

Aus den Erfahrungen und Vorteilen der komponentenbasierten Softwareentwicklung (McClure 1997; Lim 1998; Tracz 1995; Szyperski 1997) entsteht der naheliegende Wunsch, auch eine dedizierte Komponententechnik für Entwicklung und Evolution von Web-Anwendungen zu nutzen, um die Vorteile wiederverwendungsorientierter Softwareprozesse und Techniken der modernen Softwaretechnik auf die Web-Technologie übertragen zu können. Dies impliziert, neben einer adäquaten Komponententechnologie für das Web, auch insbesondere ein Vorgehensmodell im Sinne eines Softwareentwicklungsmodells für die komponentenbasierte Konstruktion und Evolution von Web-Anwendungen unter Berücksichtigung der Grundprinzipien des Webs.

Im folgenden Abschnitt dieses Beitrags werden bestehende Vorgehensmodelle zur Entwicklung von Web-Anwendungen diskutiert. Im dritten Abschnitt wird ein komponentenbasiertes Vorgehensmodell vorgestellt, daß zum einen die Wiederverwendung von Komponenten berücksichtigt und zum anderen den Grundprinzipien des Webs entspricht. Es modelliert die Evolution einer Web-Anwendung auf Basis von dedizierten Fachkomponenten. Der vierte Abschnitt skizziert kurz ein reales Anwendungssystem, das nach dem WebComposition Vorgehensmodell und mit Hilfe der WCML Komponententechnologie erstellt wurde und weiterentwickelt wird. Der Beitrag schließt mit einer kurzen Zusammenfassung.

2 Ansätze zur Softwarewiederverwendung im WebEngineering

Für die disziplinierte und wiederverwendungsorientierte Entwicklung von Web-Anwendungen ist eine ingenieurmäßige Vorgehensweise erforderlich. In diesem Abschnitt werden daher sowohl Vorgehensmodelle (*Process Models, Softwareentwicklungsmodelle*) für die Softwareentwicklung als auch Modelle für die Wiederverwendung betrachtet.

Unter einem *Vorgehensmodell* soll dabei eine allgemeine Definition von Schritten und ihrer Ausführungsreihenfolge zur Lösung von Aufgaben des gleichen Typs verstanden werden. Ein *Wiederverwendungsmodell* stellt ein besonderes Vorgehensmodell dar, das als Aufgabe Wiederverwendung von Software hat.

2.1 Vorgehensmodelle in der Softwaretechnik

Die bekannten klassischen Vorgehensmodelle, wie etwa das *Wasserfallmodell*, *explorative Vorgehensmodelle* (Sommerville 1982), *Prototypmodell* und *Spiralmodell* (Boehm 1988), lassen sich nicht, oder nur sehr eingeschränkt für die Entwicklung von Web-Anwendungen

verwenden (Powell/Jones/Cutts 1998; Lowe/Hall 1999). Probleme bereiten insbesondere die hohe Änderungsdynamik von Web-Anwendungen, die trotz oftmals hoher Anwendungsgröße und Lebensdauer auftritt, die meist starke Verteilung nicht nur der Anwendung sondern auch der Entwicklung, und letztlich die erwähnten Grundprinzipien des Web.

Seit Ende der achtziger Jahre wurden, ausgehend von den klassischen Vorgehensmodellen, mehrere neue Modelle, die einen besonderen Schwerpunkt auf die objektorientierte Entwicklung von Softwaresystemen legen, vorgestellt. Objektorientierte Entwicklung wendet sich von den funktionsorientierten Ansätzen ab, wie sie zuvor z.B. von Yourdon (Yourdon 1989) oder DeMarco (DeMarco 1978) vorgeschlagen wurden. Systeme, die durch Dekomposition der zu erzielenden Funktionalität entwickelt werden, unterliegen zumeist großen Änderungen, wenn sich ihre Anforderungen ändern. Durch die Objektorientierung kann hingegen, mit einem entsprechenden Spezifikationsmodell, eine bessere Durchgängigkeit zwischen den einzelnen Schritten während des Prozesses erzielt werden, was ein iteratives Vorgehen ermöglicht.

Bekannte Vertreter dieser Klasse von Vorgehensmodellen sind z.B. der *Semantic Object Modeling Approach* (Graham 1995), der *Objectory Software Development Process* bzw. *Unified Software Development Process* (Jacobson/Booch/Rumbaugh 1998) und der *OPEN Process* (Graham/Henderson-Sellers/Younessi 1997). Leider findet in keinem der genannten Vorgehensmodelle die explizite Unterstützung der Wiederverwendung Berücksichtigung (McClure 1997; Lim 1998; Tracz 1995); die Grundprinzipien des Webs und die Betrachtung von Software als Komponentensystem sind darüber hinaus ein weiteres Hindernis, da in diesem Fall heterogene, orthogonale Prozesse (für die unterschiedlichen Entwicklungen von Komponenten in heterogenen Umgebungen) berücksichtigt werden müßten.

Zahlreiche Forschungen im Umfeld der komponentenbasierten Softwareentwicklung wurden durchgeführt, um die Wiederverwendung in die Vorgehensmodelle zu integrieren (Sametinger 1997; Lim 1998; McClure 1997; Kang et al. 1990; Payton 1993). So schlägt etwa Ruben Prieto-Diaz zur Unterstützung der Wiederverwendung in (Biggerstaff/Perlis 1989) ein sehr allgemeines Wiederverwendungsmodell vor. Das Modell sieht drei Phasen vor, die für die Wiederverwendung notwendig sind: Zugriff auf existierenden Code (*Accessing*), Verstehen des verfügbaren Codes (*Understanding*) und Anpassen des Codes (*Adapting*). Das Vorgehen geht hierbei nicht explizit davon aus, daß zu einem früheren Zeitpunkt der Code speziell für die Wiederverwendung entwickelt wurde.

2.2 Dedizierte Vorgehensmodelle für das Web

Hypertext Design Model

Das Hypertext Design Model (HDM) (Garzotto/Paolini/Schwabe 1991; Garzotto/Mainetti/Paolini 1995) stellt ein Modell zum strukturierten Entwurf von Hypertext-Anwendungen bereit. Damit beschreibt es kein Vorgehensmodell, sondern ein Entwurfsmodell, unterstützt aber die Entwurfsphase für Hypermedia-Anwendungen eines Vorgehensmodells und läßt sich in klassische Vorgehensmodelle einbinden. Voraussetzung für die Nutzung von HDM ist, daß die Anwendung eine einheitliche und vorhersagbare Leseumgebung (*predictable reading environment*) nutzt. Der Entwurf wird hierzu in die zwei Bereiche Entwicklung im Großen (Navigationsstrukturen) und im Kleinen (einzelne Dokumente) gegliedert.

Für den Entwurf im Großen wird ein systemunabhängiges Hypertext-Modell zur Verfügung gestellt. Die Abbildung auf das Implementierungsmodell geschieht durch Instanziierung eines abstrakt erstellten Entwurfsschemas.

Die Zerlegung von Hypertext-Anwendungen in fein-granulare Artefakte für die Wiederverwendung unter Berücksichtigung sich daraus ergebener struktureller Verknüpfungen ist ein

Vorteil von HDM. Die Anwendbarkeit wird jedoch durch eine mangelnde Methodik stark eingeschränkt. Der Entwickler ist einer hohen kognitiven Belastung (Richartz 1995) ausgesetzt, die durch ein mangelndes Vorgehensmodell hervorgerufen wird, insbesondere gibt es keine Unterstützung um Artefakte wiederverwendbar zu machen. HDM ermöglicht auch nicht die Modellierung der Artefakte nach dem objektorientierten Paradigma. Die Abbildung einer in HDM entworfenen Hypertext-Anwendung auf eine Web-Anwendung wird ebenfalls durch eine mangelnde Systematik erschwert.

JESSICA

Das Projekt JESSICA (Barta/Schranz 1998) versucht den gesamten Lebenszyklus einer Web-Anwendung von der Analyse, über Entwurf und Implementierung bis hin zur Wartung abzudecken. Schranz et al. (Schranz/Weidl/Zechmeister 2000) schlagen hierzu eine objektorientierte Vorgehensweise vor, die sich an den Vorgehensmodellen der Object Oriented Analysis and Design von Yourdon (Yourdon 1994) und der Object Oriented Modeling Technique (OMT) von Rumbaugh et al. (Rumbaugh et al. 1991) orientiert. Für die Analyse wird das Konzept des Anwendungsfalls (*use case*) genutzt. Für die Notation der Ergebnisse während der Analyse und des Entwurfs wird die Unified Modelling Language (UML) verwendet. In diesen Phasen können Analyse- und Entwurfsmuster angewandt werden, um so den Prozeß zu beschleunigen.

Das JESSICA System stellt für die Modellierung eigens eine auf der Extended Markup Language (XML) basierende Sprache zur Verfügung und bildet das erstellte Modell automatisch auf Web-Ressourcen ab. Die in der JESSICA Sprache beschriebenen Entwurfsentitäten stehen den ganzen Lebenszyklus für das Management und die Wartung zur Verfügung. Die JESSICA Sprache stellt einige objektorientierte Konzepte für die abstrakte Beschreibung von Entitäten der Hypertext Markup Language (HTML) zur Verfügung. Durch Templates und JESSICA Objekte werden die Konzepte Abstraktion, Kapselung, Aggregation und Vererbung zur Verfügung gestellt. Referenzen zwischen entsprechenden Objekten werden durch das JESSICA System aufgelöst und in HTML-Links umgesetzt.

JESSICA Objekte sind lediglich Mengen von Attributen. Viele der Konzepte und Notationsmöglichkeiten der UML können daher nicht ohne Weiteres genutzt werden. Methodenaufrufe werden nicht unterstützt, da keine ausführbaren Einheiten existieren. Der Entwurf wird durch einen in seiner Funktionalität eingeschränkten und an JESSICA angepaßten UML Editor unterstützt. Eine explizite Berücksichtigung der Wiederverwendung ist nicht Bestandteil der JESSICA Methode. JESSICA Objekte können allerdings in heterogenen Umgebungen wiederverwendet werden, da sie in XML beschrieben sind. Durch die Nutzung der klassischen Vorgehensmodelle entfällt eine gesonderte Betrachtung der Wartung von Anwendungen.

Object-Oriented Hypermedia Design Method

Die Object-Oriented Hypermedia Design Method (OOHDM) (Schwabe/Rossi/Barbosa 1996) bietet im Gegensatz zu HDM eine klare Vorgehensweise für die Entwicklung von Hypermedia-Anwendungen. Das Vorgehensmodell von OOHDM besteht aus den vier Phasen Conceptual Design, Navigational Design, Abstract Interface Design und Implementierung, die nach einem iterativen, inkrementellen prototyp-basierten Vorgehensmodell durchzuführen sind.

Eine Web-Anwendung wird durch drei Modelle beschrieben (Schwabe/Rossi 1998). Das *Conceptual Model* entspricht einem herkömmlichen objektorientierten Modell, und beschreibt die Entwurfsentitäten in UML Notation, das *Navigation Model* beschreibt die Navigations-sicht auf das Conceptual Model und das *Abstract Interface Model* beschreibt die Darstellung von Schnittstellenobjekten.

Die Modularität und Wiederverwendbarkeit von Entwurfskonzepten ist durch den hohen Abstraktionsgrad der entstehenden Modelle recht hoch. Allerdings führt die Allgemeinheit der Modellierung eher zu einer Komplexitätserhöhung, so wird z.B. die Nutzung von Entwurfsmustern unterstützt, jedoch bietet die Methode keine Unterstützung bei der Suche nach anwendbaren Mustern oder bei der Automatisierung der Implementierung von wiederverwendeten Artefakten. Insgesamt betrachtet OOHDM einige wichtige Aspekte von Web-Anwendungen, es fehlt aber an einer den Grundprinzipien des Webs entsprechenden Systemunterstützung. So bleiben viele Fragen offen, wie z.B. die Implementierung in heterogenen Systemen, die Integration fremder Objekte oder das zur Verfügung stellen von Artefakten.

Relationship Management Method / RMCASE

Die Relationship Management Method (RMM) von Isakowitz et al. (Isakowitz/Stohr/Balasubramanian 1995) ist ein plattform-unabhängiges Vorgehensmodell für Hypermedia-Anwendung, das auch für die Entwicklung von Web-Anwendungen genutzt werden kann. Das Vorgehensmodell zeichnet sich durch sieben detaillierte Phasen aus. Wesentlich für die Nutzung dieses Vorgehensmodells ist die Unterstützung durch ein Werkzeug, das die Modelldaten der unterschiedlichen Phasen verwaltet, so daß die Gesamtkomplexität für den Entwickler gemindert wird.

Das RMCASE (Díaz et al. 1995) ist ein CASE-Werkzeug, das den gesamten Lebenszyklus einer Web-Anwendung unterstützt. RMCASE gliedert sich in eine Reihe von *Kontexten*, die verschiedene Sichten auf Objekte des Designs ermöglichen. Einige dieser Sichten entsprechen den Modellen einzelner Phasen des Vorgehensmodells. Für den Entwurf wird eine Sprache zur Verfügung gestellt, die es erlaubt ein an Hypertext-Systeme angepaßtes Datenmodell zu definieren, das nicht für alle Arten von Web-Anwendungen gleichermaßen genutzt werden kann, aber gut für die Abbildung von Inhalten relationaler Datenbanken geeignet ist.

Leider bleibt die Frage der Integration von Verarbeitungsprozessen ungeklärt. Der Wartungsaspekt und die Berücksichtigung der Wiederverwendung bezieht sich bedingt durch das Modell eher auf Daten als auf Artefakte bzw. Komponenten und ist auch dann nur innerhalb bestimmter einmal festgelegter Strukturen möglich. Die Wiederverwendung von Artefakten in anderen Prozessen wird nicht explizit ermöglicht.

WSDM

Die Web Site Design Method (WSDM) (De Troyer/Leune 1998) fokussiert den benutzerzentrierten anstelle des datenzentrierten Entwurfs. Der Entwurf wird ausgehend vom Blickpunkt verschiedener Benutzerklassen bestimmt und nicht von den zur Verfügung stehenden Daten.

Das Vorgehensmodell ist beschränkt auf eine Klasse reiner Informationssysteme, sogenannte „Kiosk Web-Sites“, und bietet keine Unterstützung für die Benutzerklasse „Applikations Web-Sites“, hinter der sich die restlichen, oftmals komplexeren Anwendungssysteme verbergen. Typische Problemfelder, wie z.B. Wartung, werden nicht explizit berücksichtigt. Wie bei anderen Modellen findet auch hier keine direkte Unterstützung für den Implementierungsschritt statt.

Vergleich der betrachteten Vorgehensmodelle

In Tabelle 1 werden die wichtigsten Merkmale der betrachteten dedizierten Vorgehensmodelle für das Web kurz zusammengefaßt. Als Kriterien dienen:

- Durchgängigkeit: Dieses Kriterium beschreibt die Einfachheit der Überführung von Entitäten eines Modells einer Prozeßphase in ein Modell einer vorhergehenden oder anschließenden Prozeßphase.

- **Modellmächtigkeit:** Die Modellmächtigkeit gibt an, ob das Vorgehensmodell geeignet ist, Web-Anwendungen unterschiedlichen Typs und Ausmaßes einheitlich zu beschreiben.
- **Prinzipientreue:** Hierunter soll verstanden werden, ob das Vorgehensmodell den Grundprinzipien des Webs gerecht wird.
- **Wiederverwendbarkeit:** Dieses Kriterium zeigt an, ob Entitäten der verwendeten Modelle eines Vorgehensmodells wiederverwendbar sind.
- **Wiederverwendungsunterstützung:** Dieses Kriterium dient zur Berücksichtigung expliziter Unterstützung der Wiederverwendung von Entitäten im Vorgehensmodell.

	Durchgängigkeit	Modellmächtigkeit	Prinzipientreue	Wiederverwendbarkeit	Wiederverwendungsunterstützung
HDM/HDM2	-	O	-	-	-
OOHDM	++	+	O	O	-
RMM/RMCCase	+	-	O	-	-
JESSICA	+	+	++	+	O
WSDM	--	--	-	--	--

Tabelle 1: Dedizierte Vorgehensmodelle für das Web im Vergleich

3 WebComposition Ansatz

Im folgenden wird der WebComposition Ansatz bestehend aus dem WebComposition Vorgehensmodell, Wiederverwendungsmanagement, einer Komponententechnologie und einem Evolutionskonzept basierend auf Fachkomponenten beschrieben.

3.1 WebComposition Process Model

Das WebComposition Vorgehensmodell (*WebComposition Process Model*) gliedert sich in mehrere Phasen. Die Phasen ergeben sich aus den üblichen Phasen moderner (objektorientierter) Vorgehensmodelle sowie der Umsetzung der Forderung nach Wiederverwendung. Bei dem WebComposition Vorgehensmodell handelt es sich um ein offenes Vorgehensmodell. Die Bedeutung der Offenheit wird in den folgenden Abschnitten verdeutlicht.

Beispielsweise könnte ein Entwicklungsteam eine Komponente nach dem Wasserfallmodell entwickeln, während ein anderes Entwicklungsteam ein anderes Vorgehensmodell für seinen Aufgabenbereich favorisiert. Bild 2 zeigt die Koordination mehrerer Vorgehensmodelle durch das WebComposition Vorgehensmodell und verdeutlicht die Offenheit des Vorgehensmodells für unterschiedliche Prozesse, die wiederum Instanzen unterschiedlicher Vorgehensmodelle sein können.

Die Koordination der Modelle wird durch ein explizites und koordiniertes Wiederverwendungsmanagement ermöglicht (Gaedke/Rehse/Graef 1999). Alle Vorgehensmodelle müssen daher an das Wiederverwendungsmanagement adaptiert werden, was z.B. durch Generalisierung der Vorgehensmodelle geschehen kann (McClure 1997). Dieser scheinbare Nachteil der Generalisierung von Vorgehensmodellen betrifft lediglich die Speicherung und Verwaltung von Artefakten und fällt somit bei der Anwendung eines Vorgehensmodells ohnehin an.

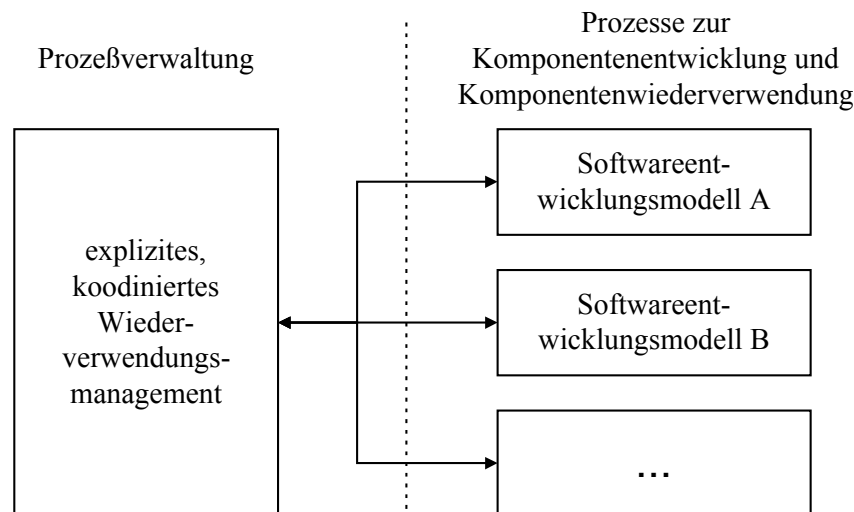


Bild 2: WebComposition Vorgehensmodell - mit unterschiedlichen Modellen

In Bild 3 wird die Sichtweise auf das Vorgehensmodell für das Entwicklungsteam mit dem Wasserfallmodell konkretisiert. Die Phasen sind plakativ an das WebComposition Vorgehensmodell adaptiert.

Die bidirektionalen Pfeile, die die Adaption an das Wiederverwendungsmanagement symbolisieren, stellen den Austausch von Artefakten des Wasserfallmodells dar. Durch die Adaption eines Modells ist das Wiederverwendungsmanagement im Besitz aller wichtigen Dokumente, Komponenten, Codefragmente, etc., die im betrachteten Prozeß bzw. im Lebenslauf der betrachteten Komponente eine Rolle spielen. Durch die Integration aller Prozesse der unterschiedlichen Vorgehensmodelle im WebComposition Vorgehensmodell kann im Wiederverwendungsmanagement über den Zustand aller Komponenten einer Komponentensoftware bzw. Web-Anwendung eine Aussage gemacht werden, die insbesondere für Wartungs- und Evolutionszwecke genutzt werden kann.

Die Durchgängigkeit bei objektorientierten Vorgehensmodellen basiert darauf, daß ein Modell in allen Phasen gemeinsam genutzt wird. Diese grundlegende Idee wird im WebComposition Vorgehensmodell aufgegriffen und auf die Koordination unterschiedlicher Prozesse von Komponentensoftware erweitert. Voraussetzung hierzu ist daher die Definition eines Modells, das sowohl für die unterschiedlichen Vorgehensmodelle, insbesondere die objektorientierten Vorgehensmodelle, als auch für die Koordination der Artefakte der einzelnen Phasen aller Prozesse geeignet ist. Es liegt somit nahe, ein objektorientiertes Modell für die Adaption im WebComposition Vorgehensmodell zu benutzen. Dieses Adaptionsmodell sowie das Wiederverwendungsmanagement werden in den anschließenden Abschnitten beschrieben, im folgenden wird daher bei der Beschreibung des WebComposition Vorgehensmodells von ihrer Existenz ausgegangen.

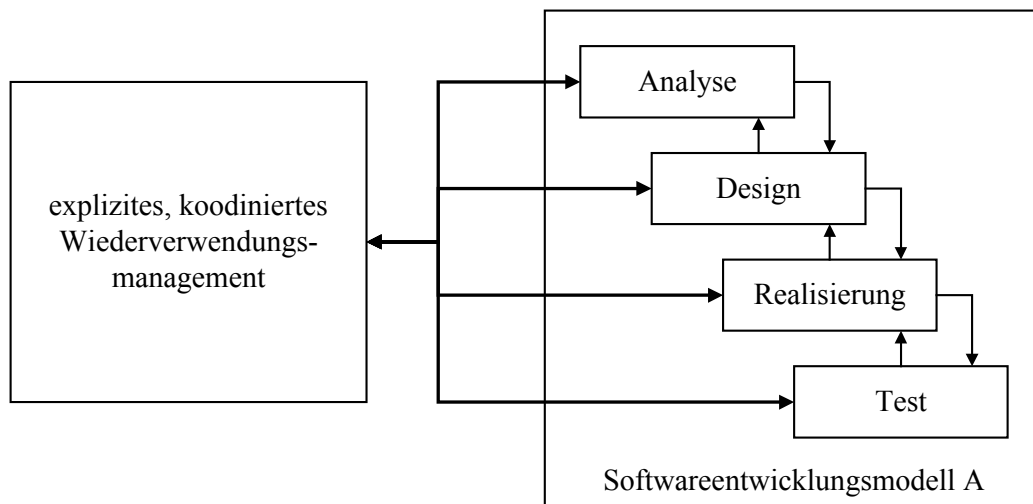


Bild 3: Wasserfall Adaption im WebComposition Vorgehensmodell

Die Anwendung eines Vorgehensmodells zur Produktion von Softwaresystemen impliziert die Erzeugung unterschiedlicher Artefakte, die als Kriterien für den Übergang von einer Stufe in die nächste vorausgesetzt werden. Betrachtet man dieses Vorgehen aus dem Blickwinkel von Komponenten und der Wiederverwendung ist es offensichtlich, alle anfallenden Artefakte als Komponenten zu betrachten und ihre „Erzeugung“ im Idealfall durch Wiederverwendung zu simulieren. Beispiele für wiederverwertbare Artefakte bzw. Komponenten sind etwa Projektplan und Pflichtenheft nach der Analysephase, Entwurfsmuster (Gamma et al. 1995), bei der Implementierung entstandene Codefragmente oder Codier- und Testrichtlinien, die in der Implementierungsphase bzw. der Testphase Verwendung finden. Ein Komponentensystem entsteht damit durch die Komposition dieser Komponenten.

Dynamische Aspekte im Vorgehensmodell

Für die Integration eines Vorgehensmodells in das WebComposition Modell müssen die während des Prozesses anfallenden Artefakte dem WebComposition Vorgehensmodell entsprechen. Hierzu müssen die Artefakte an das Modell zur Beschreibung von Artefakten im WebComposition Vorgehensmodell adaptiert werden. Nur durch die Beschreibung aller Artefakte auf Basis eines einheitlichen Modells können diese in anderen Prozessen unterschiedlicher Vorgehensmodelle wiederverwendet werden. Die Benutzung eines einheitlichen Modells erfordert in diesem Fall maximal zwei Transformationen eines Artefakts. Zuerst erfolgt die Transformation des Artefakts des Erzeuger-Prozesses in das WebComposition Modell von wo aus es zu einem späteren Zeitpunkt in ein Artefakt eines anderen Prozesses überführt werden kann. Für die Beschreibung aller Artefaktformen in einem Modell eignet sich daher ein objektorientierter Ansatz.

Aus Sicht der Wiederverwendung müssen alle Phasen demnach an geeigneter Stelle mit der Fragestellung beginnen, ob eine adäquate Lösung bereits existiert und, eventuell modifiziert, wiederverwendet werden kann. Als Beispiel soll hierfür die Entwicklung einer Bestellannahme für Mobiltelefone vorgestellt werden. In der Analyse wird festgestellt, daß das entsprechende Programm nicht existiert und zwei Schwerpunkte aufweist. Zum Einen muß die Erfassung der Kundendaten zum Anderen die Auswahl des zu bestellenden Telefons entworfen werden. Im folgenden Schritt wird festgestellt, daß für die Erfassung der Kundendaten bereits eine passende Komponente existiert. Die Entwicklung der neuen Web-Anwendung bzw. der neuen Komponente beinhaltet deshalb die Wiederverwendung der bereits existierenden Komponente zur Kundendatenerfassung. Bild 4 illustriert den Prozess und stellt zugleich den Einfluß durch die Wiederverwendung dar.

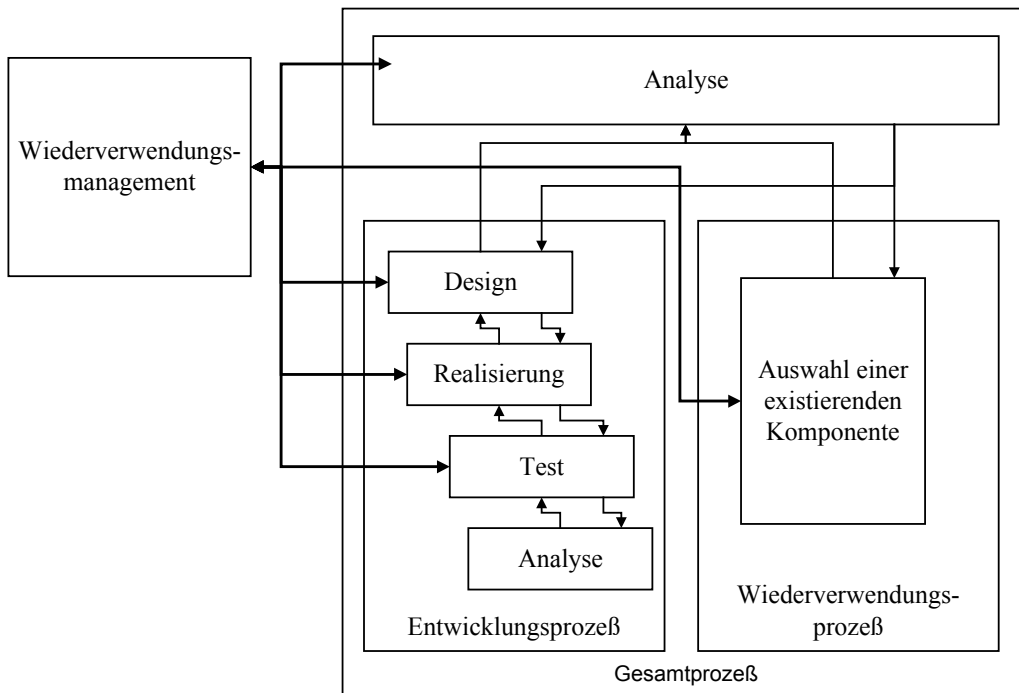


Bild 4: Aufteilung des Entwicklungsprozesses in Teilprozesse

Es ist offensichtlich, daß das WebComposition Vorgehensmodell von einer standardisierten Form von Artefakten bzw. Komponenten geprägt ist. Durch die offene und im Wiederverwendungsmanagement standardisierte Schnittstelle können beliebige Vorgehenmodelle integriert werden, wodurch der Forderung nach den Grundprinzipien des Webs nachgekommen wird.

Immanenter Aspekt im Vorgehensmodell

Der immanente Aspekt des WebComposition Vorgehensmodells zielt auf das Management und die Unterstützung der Wiederverwendung. Hierzu müssen, wie oben beschrieben, die verschiedenen Prozesse koordiniert werden, was sich auf die Verwaltung von Artefakten im WebComposition Modell zurückführen läßt. Die Schnittstelle zwischen dem Wiederverwendungsmanagement und einem Prozeß ist durch das Bearbeiten von Artefakten bzw. Komponenten gegeben. In Abbildung 5 wird das Wiederverwendungsmanagement entsprechend dem WebComposition Vorgehensmodell konkretisiert.



Abbildung 5: Immanenter Aspekt im Wiederverwendungsmanagement

Das Wiederverwendungsmodell ist an die Vorgehensmodelle durch einen Nachrichtenmechanismus zum Austausch von Komponenten und Wissen über Komponenten angebunden. Die Nachrichten können auch als Komponenten modelliert werden, so daß das Komponentenmodell auch hier durchgängig eingesetzt werden kann. Die Wiederverwendung durch das Konzept des Nachrichtenaustauschs zu realisieren, entspricht darüber hinaus dem Verteilungsaspekt des Webs. Es ist daher auch möglich, daß die Prozesse selbst verteilt ablaufen. Das Verfahren zur Realisierung des Wiederverwendungsmanagement sowie die konkrete Umsetzung der Komponententechnologie selbst ist bisher noch nicht konkretisiert worden, um die Forderung nach Offenheit und die Grundprinzipien des Webs nicht zu verletzen. Der Abschluß dieses Kapitels beschäftigt sich mit Konkretisierung und Umsetzung des WebComposition Modells sowie des Management.

3.2 Wiederverwendungsmanagement

In diesem Abschnitt wird ein Wiederverwendungsmodell vorgestellt, welches das WebComposition Vorgehensmodell konkretisiert. Ausgangspunkt ist das sogenannte Komponentendilemma (Henninger 1994a), das den Sachverhalt beschreibt, daß zwar die Wahrscheinlichkeit im Besitz einer zu einem Problem passenden Komponente zu sein mit der Anzahl der vorhandenen Komponenten steigt, aber gleichzeitig auch der Aufwand größer wird diese passende Komponente in der Menge der vorhandenen Komponenten zu lokalisieren. Das Finden von Komponenten in Bibliotheken ist deshalb ein viel betrachtetes Problem (Prieto-Díaz 1989; Prieto-Díaz 1991; Ostertag et al. 1992; Henninger 1994b).

Trennung von Modell und Repräsentation

Repräsentationen versorgen eine suchende Instanz mit Informationen über Komponenten oder liefern Klassifikationen oder Gruppierungen von Komponenten. Repräsentationen tragen also zur technischen Lösung des Problems „Finden von Komponenten“ bei (Frakes/Pole 1994; Prieto-Díaz 1991). Frakes und Pole haben den Nutzen von unterschiedlichen Repräsentationen im Rahmen einer empirischen Studie analysiert (Frakes/Pole 1994). Hierbei wurde u.a. untersucht, ob eine Repräsentation effektiver oder effizienter als andere ist. Das zentrale Ergebnis der Studie war, daß jede Repräsentation von den Teilnehmern sowohl als Beste als auch als Schlechteste für unterschiedliche Aufgaben bewertet wurde. Keine Repräsentation wurde als adäquat für alle Aufgaben angesehen. Die Studie zieht das Fazit, daß so viele unterschiedliche Repräsentationen zu einer Komponente gespeichert werden sollten wie möglich, da unterschiedliche Benutzer (als Wiederverwender von Komponenten) verschiedene Repräsentationen bevorzugen. Um dies zu erreichen ist eine Trennung von Modell und Repräsentation notwendig.

Die Trennung von einem Modell und der Sichtweise auf das Modell ist ein bekanntes Entwurfsmuster (Gamma et al. 1995). Im Fall der Repräsentation handelt es sich um die Darstellung von Informationen über Komponenten, die daher auch als Metadaten bezeichnet werden. Somit muß im strengen Sinne von mindestens zwei Modellen ausgegangen werden, einem Modell für Komponenten und einem Modell für Metadaten. Eine Repräsentation kann dann ein Metadatenmodell nutzen, um ein Ergebnis aus dem Komponentenmodell zur Verfügung zu stellen.

Die Forderung nach beliebigen Repräsentationen impliziert, daß es mehrere Metadatenmodelle mit dazugehörigen Repräsentationen geben kann aber auch, daß mehrere Repräsentationen die gleichen Metadaten nutzen können. Daher sieht die Lösung hier ein dynamisches Konzept vor, das eine lose Kopplung der Modelle und Repräsentationen zuläßt. Ausgehend von den bisher verwendeten, grundprinzipiengetreuen Konzepten kann diese Kopplung als ein Nachrichtenaustausch zwischen Metadatenmodell und Repräsentation betrachtet werden; dieser Ansatz wird in den folgenden Abschnitten genutzt.

Die grundlegende Idee hierbei ist es, lediglich die minimalen Voraussetzungen, die für die Zusammenarbeit zwischen Metadaten, Komponenten und Repräsentationsmittel notwendig sind, zu nutzen. Hierbei müssen insbesondere die Grundprinzipien des Webs beachtet werden, um den unterschiedlichen Ausprägungen von Komponenten gerecht zu werden. Die gesuchte minimale Voraussetzung ist die eindeutige Identifizierbarkeit, die Bestandteil der Definition von Komponente ist und damit auch immer erfüllt ist.

Wiederverwendungsmodell

Die beiden Betrachtungsweisen der Wiederverwendung, die des Verbrauchers (*Consumer Reuse; Development with reuse*) und des Herstellers (*Producer Reuse; Development for reuse*), wird in einem durchgängigen Vorgehensmodell integriert, das den vorgegebenen Schnittstellen des WebComposition Vorgehensmodells entspricht.

Ausgangspunkt für die Konzeption des Wiederverwendungsmodell ist das eingangs erwähnte Vorgehensmodell von Prieto-Díaz, da es durch die allgemeine Beschreibung des Wiederverwendungsmodells gut geeignet ist, auf die Aufgaben im Web Engineering konkretisiert zu werden. Der Nachteil in diesem Vorgehensmodell liegt jedoch darin, daß von einer konkreten Repräsentation ausgegangen wird. Im folgenden wird das neue Wiederverwendungsmodell durch Generalisierung des Wiederverwendungsmodells von Prieto-Díaz kurz erläutert und anschließend für die Anwendung im WebComposition Vorgehensmodell konkretisiert. Die notwendige Systemunterstützung wird durch ein Reuse-Repository ermöglicht, dessen Beschreibung in (Gaedke/Rehse/Graef 1999) zu finden ist.

Wiederverwendungsmodell aus Sicht des Verbrauchers

Die Generalisierung des Wiederverwendungsmodells findet durch die Verallgemeinerung der Spezifikation statt, d.h. die Art der Spezifikation wird nicht mehr vorgegeben. Während im ursprünglichen Wiederverwendungsmodell die Funktion einer Komponente spezifiziert werden mußte, wird diese Anforderung durch eine undefinierte aber wählbare Eigenschaft ersetzt. Damit lassen sich im neuen Wiederverwendungsmodell beliebige Eigenschaften für die Auswahl einer Komponente heranziehen, ohne dabei das Wiederverwendungsmodell zu verletzen. Das modifizierte Wiederverwendungsmodell setzt für die Auswahl von Komponenten die Existenz einer Menge von Komponenten voraus. Die Organisation dieser Menge wird hierbei nicht weiter betrachtet, so daß das Wiederverwendungsmodell keine Einschränkungen für eine bestimmte Komponententechnologie darstellt. Lediglich die Eindeutigkeit bzw. Identifizierbarkeit, wie sie in der Definition von Komponente gefordert wird, wird genutzt.

Henninger beschreibt in (Henninger 1994b), daß zu Beginn des Wiederverwendungsprozesses häufig keine vollständige und qualitativ ausreichende Anfrage formuliert werden kann, die zu der gewünschten Komponente führt. Das iterative Vorgehen durch Re-Spezifikation der Suchanfrage, wie es auch häufig von großen Suchmaschinen z.B. in Bibliotheken angeboten wird, hat sich diesbezüglich als vorteilhaft erwiesen. Ein potentieller Wiederverwender kann das durch die Anfrage erworbene Wissen nutzen, um eine verfeinerte Anfrage zu spezifizieren. Während im ursprünglichen Wiederverwendungsmodell die gefundenen Komponenten der Spezifikation einem bestimmten Grad entsprechen müssen, der auch Erfüllungsgrad genannt wird, wird dies im generalisierten Wiederverwendungsmodell nicht weiter spezifiziert. Die Bestimmung der Lösungsmenge muß weder berechenbar noch nachvollziehbar sein. Dies ist eine wichtige Notwendigkeit, um beliebige Repräsentationen an das Wiederverwendungsmodell zu binden. Die folgende Abbildung zeigt ein Werkzeug, daß das Finden von Komponenten durch eine grafische Schnittstelle ermöglicht. Die Unterstützung der Wiederverwendung durch Werkzeuge, die Metadaten auswerten, Metadatenpeicher und Komponentenspeicher wird im WebComposition Vorgehensmodell auch als *Reuse Repository* bezeichnet. Ein Beispiel für eine grafische Unterstützung bei der Suche nach Komponenten zeigt das folgende Bild 6. In dem dargestellten Werkzeug ist eine Navigation durch eine Komponentenmenge

über einen Graph möglich, dessen Knoten Komponenten darstellen und dessen Kanten durch Prototyp-Instanz-Relationen zwischen Komponenten definiert sind.

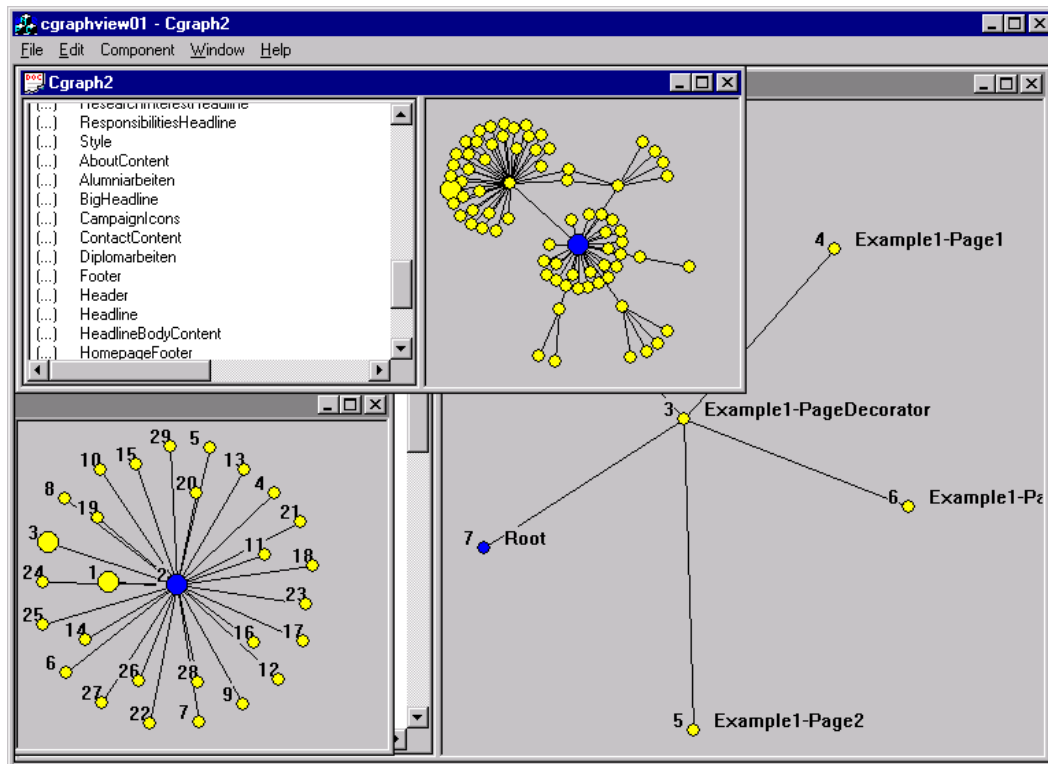


Bild 6: Grafische Unterstützung bei der Suche nach Komponenten

Allgemeines Wiederverwendungsmodell aus Sicht des Herstellers

Aus Sicht des Herstellers von Komponenten wird ein Mechanismus benötigt, um Metadaten über Komponenten anzulegen, damit die Komponente entsprechend ihrer Funktionalität klassifiziert werden kann. Dies kann durch manuelle Angabe der Attribute zur Beschreibung einer Komponente erfolgen oder durch ein automatisches Verfahren, das Metadaten aus anderen Informationsquellen (etwa Entwicklungssystem, Dokumentationssystem etc.) oder durch Analyse der Komponente selbst erzeugt. Die Automatisierung kann dabei prinzipiell sowohl auf syntaktischer als auch auf semantischer Ebene erfolgen. Die Daten können auch Qualitätsmerkmale oder Zertifizierungsinformationen beinhalten.

3.3 Die WebComposition Komponententechnologie

In diesem Abschnitt wird nun eine dedizierte Komponententechnologie vorgestellt, die den Ansprüchen der vorausgegangenen Modelle entspricht und die Umsetzung des WebComposition Vorgehensmodells ermöglicht.

Komponentenbasierte Artefaktabstraktion

Wie bereits deutlich wurde, ist das Hauptproblem im Web Engineering die durchgängige und wiederverwendbare Modellierung von Artefakten des Vorgehensmodells zur Entwicklung und Wartung von Web-Anwendungen. Insbesondere stellt die Lücke zwischen Entwurfsmodell und Implementierungsmodell ein besonderes Erschwernis dar (Gellersen 1997; Gaedke 1998; Barta/Schranz 1998).

Für die durchgängige Wiederverwendung von Artefakten ist daher eine komponentenorientierte Sichtweise anzustreben. Komponenten sind identifizierbar und können im Wiederver-

wendungsmodell eingesetzt werden. Ferner ist es wichtig, daß die Abbildung der Inhalte auf Ressourcen als Entwurfsentscheidung im Entwurf modelliert werden können und auch nach der Implementierung zugreifbar bleiben. Dies impliziert, daß die Elemente einer Web-Anwendung unabhängig vom Begriff der Ressource modelliert werden müssen. Objekte des Modells sollten auf Teile von Ressourcen ebenso wie auf ganze Ressourcen oder sogar Mengen von Ressourcen abgebildet werden können und müssen als autonome Elemente bzw. Komponenten anderer (orthogonaler) Prozesse nach dem WebComposition Vorgehensmodell für die Wiederverwendung zur Verfügung stehen.

WebComposition Markup Language

Die Beschreibung von Komponenten und Beziehungen zwischen Komponenten erfolgt mit Hilfe der WebComposition Markup Language (WCML) (Gaedke/Schempf/Gellersen 1999), welche selbst eine Anwendung von XML darstellt. WCML stellt Komponenten als uniformes Modellierungskonzept für Artefakte im Lebenszyklus von Web-Anwendungen zur Verfügung. Das Modell unterstützt zum einen eine automatische aber auch noch im Detail manipulierbare Abbildung auf Entitäten des Implementierungsmodells, während es andererseits auch die Modellierung beispielsweise von Analyseartefakten, Entwurfsartefakten oder Wiederverwendungsprodukten durch objektorientierte Konzepte und Abstraktionsmechanismen ermöglicht.

Eine Web-Anwendung wird im WCML-Modell als Komposition von Komponenten beschrieben. Aus der Sicht der unterschiedlichen Prozessfortschritte können daher Web-Anwendungen aus einer Hierarchie von fertiggestellten Komponenten, die dabei ganzen Teilen von Web-Anwendungen, mit einzelnen Ressourcen oder Fragmenten von Ressourcen entsprechen; andererseits existieren für bestimmte Teile einer Web-Anwendung evtl. auch nur Informationen aus der Analyse oder dem Entwurf. Die letzt genannten Komponenten sind Bestandteil der Anwendung und Hinweis dafür, wie sich die Anwendung weiterentwickeln wird oder kann. Komponenten im WCML-Modell sind durch eine „Universally Unique Id“ (UUID) eindeutig identifizierbar. Sie besitzen einen Zustand in Form von typisierten Attributen, die Property genannt werden, und einfachen Name-Wert Paaren entsprechen. Der Wert eines Property kann durch statischen Text oder WCML-Sprachkonstrukte definiert werden, dabei muß der Wert dem Datentyp des Property entsprechen.

Das schon eingangs aufgegriffene Konzept der Erstellung von Komponentensoftware durch Komposition wird durch die WCML-Sprachkonstrukte realisiert. Jede Komponente kann dabei auf beliebig viele andere Komponenten aufbauen und deren Verhalten nutzen, indem sie Komponenten referenziert oder als Prototyp verwendet. Eine Modifikation an einer Komponente kann damit eine konsistente Änderung an allen Komponenten bewirken, die sie nutzen. Für eine detailliertere Beschreibung von WCML sei auf (Gaedke/Schempf/Gellersen 1999; Gaedke 1999) verwiesen.

3.4 Evolution mit Fachkomponenten

Die Anforderungen an ein Software System ändern sich im Laufe der Zeit. Es ist offensichtlich, daß hierfür eine große Menge von Einflüssen verantwortlich sind, wie z.B. Gesetzliche Bestimmungen, eine nicht zur Zeit oder Firmenkultur adäquate Präsentation oder die Erweiterung der Funktionalität. Solche Wartungsaufgaben sind schwer zu bewerkstelligen, wenn die Anwendung nicht vorausschauend für spätere Änderungen und Erweiterungen konzipiert worden ist.

Um eine disziplinierte und planbare Evolution einer Web-Anwendung in der Zukunft zu ermöglichen, ist es sinnvoll die Initialanwendung nicht auf der Basis der zu Beginn anfallenden konkreten Anforderungen zu definieren. Vielmehr soll die Initialanwendung als leere Anwen-

ung betrachtet werden, die zur Aufnahme von Funktionalität innerhalb eines klar definierten Evolutionsraumes geeignet ist.

Eine Grundlage für diesen Ansatz stellt das Domain Engineering dar. Domain Engineering wird beschrieben, als ein Prozeß für die Erzeugung von Kompetenz auf dem Gebiet der Anwendungsentwicklung für eine Familie ähnlicher Systeme (Institute 1999). In einer Analysephase werden die Eigenschaften eines Anwendungsgebietes ermittelt und in einer Entwurfsphase in ein Modell für die Anwendungsdomäne umgesetzt. Daraus läßt sich der benötigte Evolutionsraum bestimmen und die Initialanwendung im Sinne der Implementierungsphase des Domain Engineering als Framework erstellen, das dann zur Aufnahme beliebiger Funktionalität in dem durch die Domäne abgeleiteten Evolutionsraum in der Lage ist.

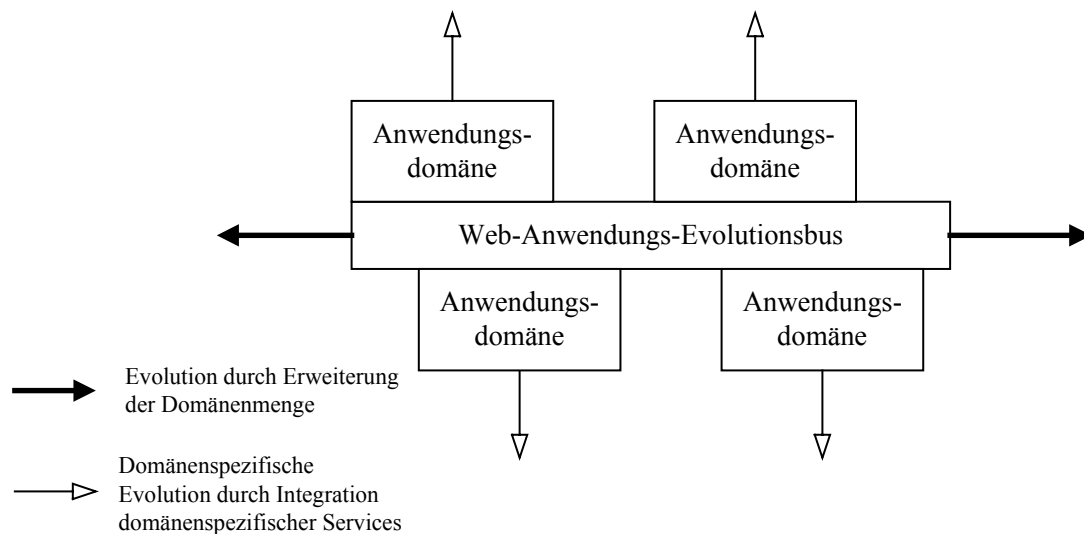


Bild 7: Dimensionen des Evolutionsraumes einer Web-Anwendung

Diese Betrachtungsweise läßt sich im Sinne des WebComposition Vorgehensmodells auf mehrere Anwendungsdomänen erweitern. Die Basisarchitektur einer Web-Anwendung bezeichnen wir deshalb im folgenden als *Evolutionsbus*. Der Evolutionsbus stellt die Initialanwendung für die gesamten abstrakten Anwendungsdomänen einer Web-Anwendung dar.

Er ermöglicht die Verwaltung und die Zusammenarbeit von Fachkomponenten, d.h. WCML Komponenten die konkrete Anwendungsdomänen umsetzen, wie z.B. Web-basiertes Procurement, Informationsdarstellung, benutzergesteuerter Informationsaustausch. Diese Fachkomponenten, die auch als Dienste (*Services*) im WebComposition Vorgehensmodell bezeichnet werden, stellen zugleich die Prototypen für zukünftige Services der gleichen Anwendungsdomäne dar. Die Evolution kann dann auf zwei definierte Arten stattfinden, vergl. auch Bild 7:

- *Domänenspezifische Evolution* – Die Erweiterung einer Domäne um Services, z.B. durch Prototyping eines existierenden Dienstes der Domäne. Eine weitere Möglichkeit ist, daß sich die Domäne selbst ändert oder mehr Funktionalität bekommt, wozu eine Modifikation des initialen Services der Domäne, der als Prototyp für andere Services fungiert, notwendig ist.
- *Evolution der Domänenmenge* – Die Evolution der Anwendung kann schließlich auch durch die Modifikation der Domänenmenge stattfinden. Die Erweiterung der Funktionalität einer Anwendung durch eine neue Anwendungsdomäne findet z.B. beim Übergang

eines Web-basierten Produktkataloges zu einem Produktkatalog mit Warenkorbfunktionalität statt. Die Integration einer neuen Domäne wird durch die Verbindung eines neuen initialen Service mit dem Evolutionsbus realisiert.

Ein Framework für die Realisierung des Evolutionsbusses und die Services lassen sich mit WCML entwickeln, wie in (Gaedke/Turowski 1999a; Gaedke/Turowski 1999b) dargestellt. Durch die verschiedenen Granularitätsstufen der Service-Komponenten lassen sich darüber hinaus Werkzeuge entwickeln, die die Evolution besonders fördern, wie z.B. Service-Factories und domänenspezifische Sprachen (Graef/Gaedke 1999).

4 Anwendung des WebComposition Vorgehensmodell

In einem Projekt zwischen dem Telecooperation Office (TecO) an der Universität Karlsruhe und der Hewlett-Packard GmbH (HP) wurde eine Service-orientierte E-Commerce Anwendung, namens Eurovictor, basierend auf dem WebComposition Vorgehensmodell umgesetzt. Ziel des Projekts war es, ein Unterstützungssystem zu entwickeln, das Entwicklung, Management, Wartung und Evolution von Services im heterogenen Umfeld des europäischen Intranets der Firma HP ermöglicht. Im Rahmen dieses Projekts wurde ein Evolutionsbus realisiert. Die Evolution der Anwendung vollzog sich durch die Integration von einigen domänenspezifischen Services, z.B. Informationsdarstellung, Softwarebestellung, Produktkauf. Diese Services dienen seither als Prototyp für die domänenspezifische Evolution der Anwendung. In der folgenden Abbildung ist die Startseite der Web-Anwendung dargestellt. Der linke Bereich der Web-Seite zeigt ein Menü, daß die Anwahl von Diensten ermöglicht. In der Mitte und auf der rechten Seite sind zwei spezielle Dienste, zur Anwendungsadaption an das Benutzerverhalten sowie ein Warenkorbdienst, dargestellt. Der Dienst zur Anwendungsadaption zeigt darüber hinaus die Flexibilität der Anwendung durch die fachkomponentenspezifische Anwendungsarchitektur.

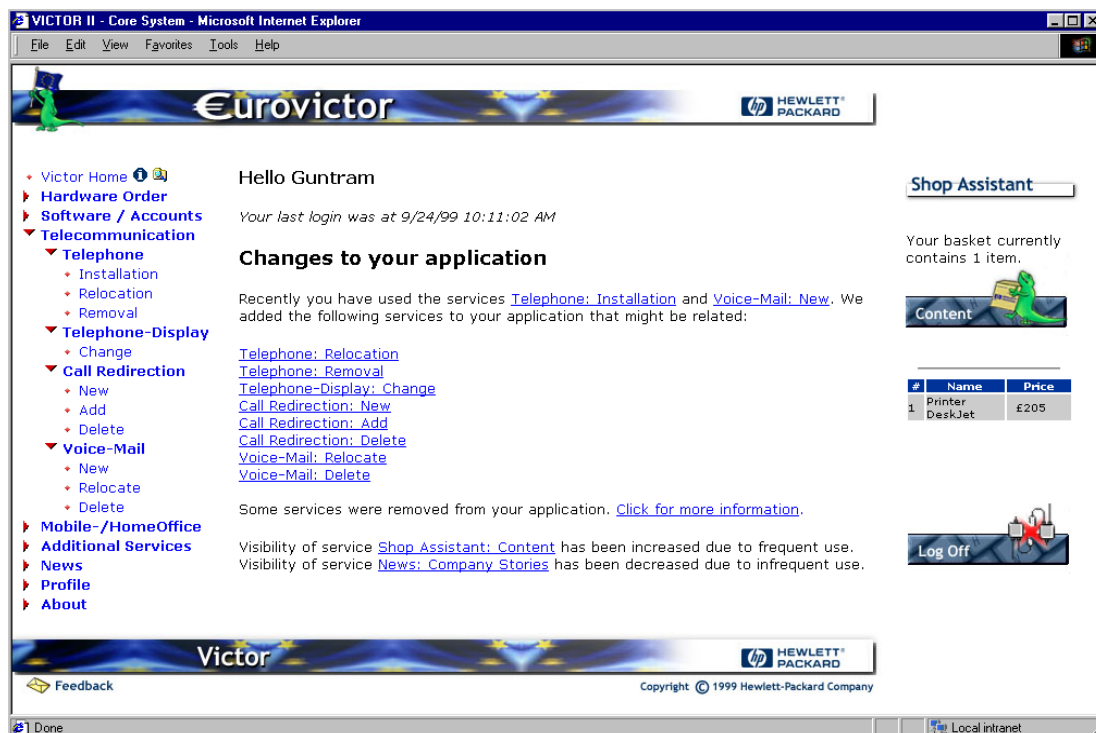


Abbildung 8: Beispielanwendung Eurovictor

Die domänenspezifischen Evolution der Anwendung findet insbesondere im internationalen Einsatz statt, so werden z.B. neue Services durch Vererbung erstellt, die an nationale Layouts, Sprachen oder gesetzliche Bestimmungen angepaßt werden. Mittlerweile werden in nahezu allen europäischen Ländern Services verteilt entwickelt und über das Eurovector System zur Verfügung gestellt.

5 Zusammenfassung

Das Web hat sich als Plattform für Anwendungen etabliert, obwohl Erstellung und Evolution von Web-Anwendungen durch das dem Web zugrunde liegende Implementierungsmodell erschwert werden. Die zunehmende Komplexität von Anwendungen im Web macht eine disziplinierte Vorgehensweise für Entwicklung und Evolution zwingend erforderlich. Die Nutzung der Vorteile komponentenbasierter Softwareentwicklung im Web Engineering ist daher wünschenswert, wird aber durch die Grundprinzipien im Web, die die autonome Entwicklung in heterogenen Umgebungen fordern, erschwert.

Das WebComposition Vorgehensmodell beschreibt die durchgängige Vorgehensweise zur Entwicklung von Web-Anwendungen als Komponentensoftware. Hierzu führt es das Konzept des offenen Vorgehensmodells ein, daß die Integration beliebiger Prozesse zur Entwicklung und Wiederverwendung von Komponenten, also Teilen der Zielanwendung, zuläßt. Die Wiederverwendung wird gefördert, indem alle im Lebenszyklus einer Komponente anfallenden Artefakte als standardisierte Komponenten modelliert werden müssen und die Nutzung von Artefakten anderer Prozesse explizit durch ein Wiederverwendungsmodell unterstützt wird. Die Artefakte werden als Komponenten in der WebComposition Markup Language modelliert, die als Anwendung der XML den Grundprinzipien des Webs entspricht. Die Evolution einer Web-Anwendung wird im WebComposition Vorgehensmodell durch das Konzept des Domain Engineering geplant. Die Anwendungsdomänen werden durch Services beschrieben, die Fachkomponenten entsprechen. Die Evolution nach Anwendungsdomänen ist zentraler Bestandteil des Vorgehensmodell und wird durch den sog. Evolutionsbus, ein Framework für die Integration von Fachkomponenten, beschrieben. Das WebComposition Vorgehensmodell wurde bereits erfolgreich in der Praxis eingesetzt. Am Beispiel einer großen, internationalen Web-Anwendung der Firma Hewlett-Packard, die nach dem Vorgehensmodell entwickelt wurde, werden die Vorzüge für die Evolution aufgezeigt.

Anwendungsbeispiele

Der WCML-Compiler und Anwendungsbeispiele sind unter <http://www.webengineering.org> zu finden.

Literatur

- Barta, R. A.; Schranz, M. W.:* JESSICA: an object-oriented hypermedia publishing processor. In: Computer Networks and ISDN Systems 30(1998) (1998) Special Issue on the 7th Intl. World-Wide Web Conference, Brisbane, Australia, S. 239-249.
- Berners-Lee, T.* Information Management: A Proposal. <http://www.w3.org/Proposal.html>, Abruf am 1998-10.10.1998.
- Biggerstaff, T. J.; Perlis, A. J.:* Software reusability. ACM Press; Addison-Wesley, New York, N.Y.; Reading, Mass. 1989.
- Boehm, B. W.:* A Spiral Model of Software Development and Enhancement. In: Computer 21 (1988) 5, S. 61-72.
- Cusumano, M. A.; Yoffie, D. B.:* Software Development on Internet Time. In: IEEE Computer 32 (1999) 10, S. 60-69.

- De Troyer, O. M. F.; Leune, C. J.:* WSDM: a user centered design method for Web sites. In: Computer Networks and ISDN Systems 30 (1998) (1998) Special Issue on the 7th Intl. World-Wide Web Conference, Brisbane, Australia.
- DeMarco, T.:* Structured analysis and system specification. Yourdon, New York 1978.
- Díaz, A.; Isakowitz, T.; Maiora, V.; Gilibert, G.:* RMC: A Tool To Design WWW Applications. In: The World Wide Web Journal 1 (1995).
- Frakes, W. B.; Pole, T. P.:* An Empirical Study of Representation Methods for Reusable Software Components. In: IEEE Transactions on Software Engineering 20 (1994) 8, S. 617.
- Gaedke, M.:* WebComposition: Ein Unterstützungssystem für das Web Engineering. In: GI Softwaretechnik-Trends 18 Heft 3 (1998) Special Issue Softwaretechnik (ST'98), Paderborn Germany.
- Gaedke, M.:* Wiederverwendung von Komponenten in Web-Anwendungen. In: *K. Turowski (Hrsg.): 1. Workshop Komponentenorientierte betriebliche Anwendungssysteme (WKBA 1).* Magdeburg, Germany 1999.
- Gaedke, M.; Rehse, J.; Graef, G.:* A Repository to facilitate Reuse in Component-Based Web Engineering. International Workshop on Web Engineering at the 8th International World-Wide Web Conference (WWW8). Toronto, Ontario, Canada 1999.
- Gaedke, M.; Schempff, D.; Gellersen, H.-W.:* WCML: An enabling technology for the reuse in object-oriented Web Engineering. Poster-Proceedings of the 8th International World Wide Web Conference (WWW8). Toronto, Ontario, Canada 1999.
- Gaedke, M.; Turowski, K.:* Generic Web-Based Federation of Business Application Systems for E-Commerce Applications. In: *S. Conrad; W. Hasselbring; G. Saake (Hrsg.): Second International Workshop on Engineering Federated Information Systems (EFIS'99).* Kühlungsborn 1999a.
- Gaedke, M.; Turowski, K.:* Integrating Web-based E-Commerce Applications with Business Application Systems. In: Netnomics, Baltzer Science Publishers under submission (1999b) Special Issue on Information and Communication Middleware, Fawzi Daoud (Ed.).
- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.:* Design patterns: elements of reusable object-oriented software. Addison-Wesley, Reading, Mass. 1995.
- Garzotto, F.; Mainetti, L.; Paolini, P.:* Hypermedia design, analysis, and evaluation issues. In: Communications of the ACM 38 (1995) 8, S. 74-86.
- Garzotto, F.; Paolini, P.; Schwabe, D.:* HDM - A model for the Design of Hypertext Applications. Hypertext'91. 1991, S. 313-326
- Gellersen, H.-W.:* Web Engineering: Softwaretechnik für Anwendungen im World-Wide Web. In: HMD, Theorie und Praxis der Wirtschaftsinformatik 36 (1997) 196.
- Gellersen, H.-W.; Gaedke, M.:* Object-Oriented Web Application Development. In: IEEE Internet Computing 3 (1999) 1, S. 60-68.
- Gellersen, H.-W.; Wicke, R.; Gaedke, M.:* WebComposition: an object-oriented support system for the Web engineering lifecycle. In: Computer Networks and ISDN Systems 29 (1997) (1997) Special Issue on the 6th Intl. World-Wide Web Conference, Santa Clara, CA, USA, S. 1429-1437.
- Graef, G.; Gaedke, M.:* An Evolution-oriented Architecture for Web Applications. In: *J. Bosch (Hrsg.): Second Nordic Workshop on Software Architecture (NOSA '99).* Bd. 99, Ronneby, Sweden 1999.
- Graham, I.:* Migrating to Object Technology. Addison-Wesley, Reding, MA 1995.
- Graham, I.; Henderson-Sellers, B.; Younessi, H.:* The OPEN Process Specification. Addison-Wesley, New York, NY 1997.
- Henninger, S.:* Supporting the Construction and Evolution of Component Repositories. 18th International Conference on Software Engineering (ICSE). Bd. 18, Berlin, Germany 1994a, henninger_evolution.
- Henninger, S.:* Using Iterative Refinement to Find Reusable Software. In: IEEE Software 94 (1994b) 5.
- Institute:* Software Engineering Institute - Domain Engineering. http://www.sei.cmu.edu/domain-engineering/domain_engineering.html, Abruf am 1999-01.10.1999.
- Isakowitz, T.; Stohr, E. A.; Balasubramanian, P.:* RMM: A Methodology for Structured Hypermedia Design. In: Communications of the ACM 38, No. 8 (1995) Aug. 1995, S. 34-44.

- Jacobson, I.; Booch, G.; Rumbaugh, J.:* The Objectory Development Process. Addison-Wesley, Reading, MA 1998.
- Kang, K.; Cohen, S.; Hess, J.; Novak, W.; Peterson, A. (1990).* Feature-Oriented Domain Analysis (FODA) Feasibility Study. Pittsburgh, PA: Carnegie-Mellon University.
- Lim, W. C.:* Managing software reuse : a comprehensive guide to strategically reengineering the organization for reusable components. Prentice Hall, Upper Saddle River, NJ 1998.
- Lowe, D.; Hall, W.:* Hypermedia & the Web : an engineering approach. John Wiley, Chichester; New York 1999.
- McClure, C. L.:* Software reuse techniques : adding reuse to the system development process. Prentice Hall, Upper Saddle River, N.J. 1997.
- Ostertag, E.; Hendler, J.; Prieto-Díaz, R.; Braun, C.:* Computing Similarity in a Reuse Library System: An AI-Based Approach. In: ACM Transactions on Software Engineering and Methodology 1 (1992) 3.
- Payton, T.* Megaprogramming - Facilitating a Transition Towards Product-Line Software. <http://source.asset.com/stars/darpa/Papers/megaprog/terimega.html>.
- Powell, T. A.; Jones, D. L.; Cutts, D. C.:* Web site engineering : beyond Web page design. Prentice Hall PTR, Upper Saddle River, NJ 1998.
- Prieto-Díaz, R.:* Classification of Reusable Modules. In: *T. J. Biggerstaff; A. J. Perlis (Hrsg.): Software Reusability.* Bd. 1, ACM Press, 1989,.
- Prieto-Díaz, R.:* Implementing Faceted Classification for Software Reuse. In: Communications of the ACM 34 (1991) 5.
- Richartz, M. (1995).* Generik und Dynamik in Hypertexten. Dissertation am Institut für Telematik, Telecooperation Office. Karlsruhe: Universität Karlsruhe.
- Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy, F.; Lorenzen, W.:* Object-Oriented Modeling and Design. Prentice Hall, Englewood Cliffs, NY 1991.
- Sametinger, J.:* Software engineering with reusable components. Springer, Berlin ; New York 1997.
- Schranz, M.; Weidl, J.; Zechmeister, J.:* Engineering Complex World Wide Web Services with JESSICA and UML. Thirty-third Annual Hawai'i International Conference on Systems Sciences (HICSS'33). Maui, HI, USA 2000
- Schwabe, D.; Rossi, G.:* An Object Oriented Approach to Web-Based Applications Design. In: TAPOS - Theory and Practice of Object Systems 4 (1998) 4, S. 207-225.
- Schwabe, D.; Rossi, G.; Barbosa, S.:* Systematic Hypermedia Design with OOHDM. ACM International Conference on Hypertext' 96. Washington, USA 1996
- Sommerville, I.:* Software Engineering. Addison-Wesley Pub. Co., London ; Reading, Mass. 1982.
- Szyperski, C.:* Component software: beyond object-oriented programming. ACM Press; Addison-Wesley, New York, Harlow, England; Reading, Mass. 1997.
- Tracz, W.:* Confessions of a used program salesman : institutionalizing software reuse. Addison-Wesley Pub. Co., Reading, Mass. 1995.
- Yourdon, E.:* Modern Structured Analysis. Yourdon Press, Englewood Cliffs, New Jersey 1989.
- Yourdon, E.:* Object-Oriented Systems Design: An Integrated Approach. Prentice-Hall, New York 1994.