

Interaktion im Web

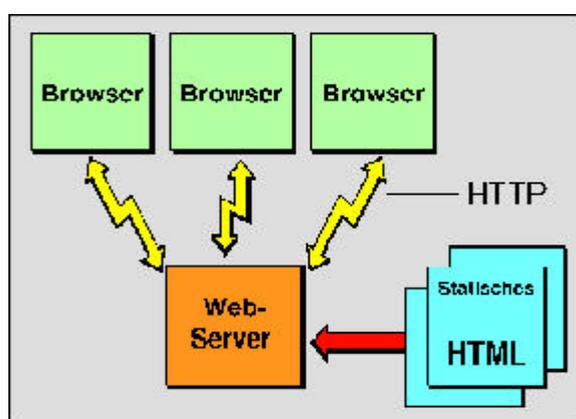
Dynamisch generierte HTML-Seiten

Das Web, der am schnellsten wachsende Teil des Internet, fordert bislang wenig zu Interaktionen zwischen Nutzern und entfernten Servern auf. Dabei bieten die an beiden Enden des Kommunikationsstranges zwangsläufig vorhandenen Rechner weitaus mehr Anwendungsmöglichkeiten als das einfache Hin- und Herblättern durch Web-Seiten. Eine Variante, um Interaktion im Web zu gestatten, ist das dynamische Generieren von HTML-Seiten mit Hilfe von WebObjects. Diese Entwicklungs- und Laufzeitumgebung stellt mittlerweile in der Version 4.0 die dazu erforderlichen Werkzeuge zur Verfügung.

Einleitung

Anwendungen im Internet finden, auch im Bauwesen, immer weitere Verbreitung. Vor allem Applikationen, die in der Lage sind, dynamisch auf Benutzer zu reagieren, sind im Aufwind. Zu ihnen zählen die mit WebObjects erstellten Programme. Dieses Entwicklerpaket soll hier in seinen Grundzügen vorgestellt werden.

handene Daten und Systeme zurückgreifen kann. WebObjects zielt daher in zwei Richtungen, einerseits auf das einfache Publizieren dynamisch generierter HTML-Seiten, andererseits auf die Weiterentwicklung klassischer Client-Server-Applikationen. Mit den ungeheuren Verbindungsmöglichkeiten des Internet eröffnen sich in diesem Anwendungsbereich ganz neue Perspektiven.



Statisches Publizieren von HTML-Seiten

Was ist WebObjects?

Als Softwareumgebung dient WebObjects der schnellen Erstellung und zum Ablauf von Server-Anwendungen im „World Wide Web“. Der Einsatzschwerpunkt liegt auf der Entwicklung dynamischer Web-Seiten und -Applikationen. Das Paket besteht aus Werkzeugen, Objektbibliotheken und einem Laufzeitsystem. Es ist standardkompatibel und arbeitet mit allen Web-Browsern und -Servern auf der Basis der gängigen Schnittstellen zusammen.

Aus der Sicht des Anwenders auf der Benutzerseite ist WebObjects einfach ein Applikations-Server. Aus der Sicht des Entwicklers auf der Serverseite ist WebObjects eine Plattform, auf der er schnell skalier- und erweiterbare Internetapplikationen entwickeln, zur Verfügung stellen und dabei auf vor-

Publizieren im Internet

Aktivitäten im Internet stellen sich heute fast ausschließlich als das Publizieren von HTML-Seiten dar. Auf ihnen werden in mehr oder minder ansprechender Form den Benutzern als Leser Inhalte zur Verfügung gestellt. Dazu wird das HTTP verwendet, ein Protokoll, das 1991 entwickelt wurde und seitdem das Internet maßgeblich geprägt hat.

Statisches HTML

Die meisten Anbieter im Web präsentieren nach wie vor statische HTML-Seiten, deren Inhalte sich über lange Zeiträume kaum oder gar nicht ändern. Eine solchermaßen typische Web-Site reagiert auf die normalerweise mit einem

Browser erzeugten Anfragen aus dem Internet, indem einfach über das Netzwerk der Web-Server aufgerufen wird. Nur dieses Programm kann die HTTP-konforme Anfrage analysieren und bearbeiten. Es fordert die gewünschte Seite aus dem lokalen Netzwerk an und sendet sie dem anfragenden Browser zu.

Die Seite ist ein einfaches Textdokument, das neben dem Inhalt in HTML kodierte Anweisungen enthält. Mit diesen eingebetteten Anweisungen kann der Browser auf der Benutzerseite die empfangene Datei bearbeiten und angemessen auf den unterschiedlichsten Bildschirmen darstellen.

Statische Seiten sind einfach herzustellen. Neben der texteditierten Erzeugung gibt es mittlerweile eine Reihe von Entwicklungswerkzeugen, mit denen selbst an-

spruchsvollste HTML-Seiten auf einfache Art und Weise erstellt werden können. Solange sich deren Inhalte nicht allzu oft ändern, sind sie auf dieselbe Weise aktuell zu halten. Sollen so allerdings mehr oder minder fortlaufend aktuelle Seiten publiziert werden, bedarf es dazu einiger Anstrengungen.

Dynamische Seiten

Demgegenüber stehen dynamisch erzeugte HTML-Seiten, die aus vorbestimmten Datenbeständen

von Rechnern automatisch erzeugt werden. Mit der Bereitstellung der geforderten HTML-Seite müssen gleichzeitig die Inhalte formuliert werden.

Bekannteste Beispiele sind sicherlich die Suchmaschinen, bei denen die Suchergebnisse in eine vorge-

faßte Schablone als Text mit der entsprechenden HTML-Kodierung eingepaßt werden. Eine andere Variante ist ein Seitenzähler (hit counter), den auch viele Privatleute auf ihre Seiten setzen. Ein weiterer wichtiger Bereich sind die Internetshops. Hier können ebenfalls nur dynamisch erzeugte HTML-Seiten zum Einsatz gelangen, da nur so die spezifischen Such- und Einkaufsfunktionen angemessen zu unterstützen sind.

Nicht zu den dynamischen HTML-Seiten zu rechnen sind dagegen die Java-Applets. Diese Programmteile werden zwar vom Server zur Verfügung gestellt, sind aber nur auf der Benutzerseite innerhalb des Browserfensters lauffä-

hig und bieten auf direktem Weg keine Interaktivität über das HTTP.

Mittlerweile werden die Programme, mit denen nichtstatische Webseiten im Internet angeboten werden, allgemein als Applikations-Server bezeichnet. Die meisten davon stellen konzeptionell Erweiterungen der spezifischen Webserver dar und bieten jeweils eigene Möglichkeiten, mittels einer Skriptsprache Applikationen zu entwickeln.

Die Unterschiede in der Herangehensweise an die Programmierung sind groß, eine angemessene Bewertung deswegen kaum möglich. Welche Konzepte es hier beispielsweise mit „Perl“ gibt, soll an dieser Stelle im Zusammenhang mit WebObjects nicht weiter ausgeführt werden.

Das Konzept von WebObjects

WebObjects verfolgt ein anderes Konzept. Ausgehend von der ursprünglichen Absicht, ein auf Unix beruhendes Betriebssystem zu vervollständigen, wurde eine bereits vorhandene Programmierumgebung weiter entwickelt. Von Anfang an lag der Schwerpunkt bei WebObject darauf, schnell, einfach und dynamisch Daten über das Internet zu transferieren und dabei vorhandene Entwicklungswerkzeuge zu nutzen.

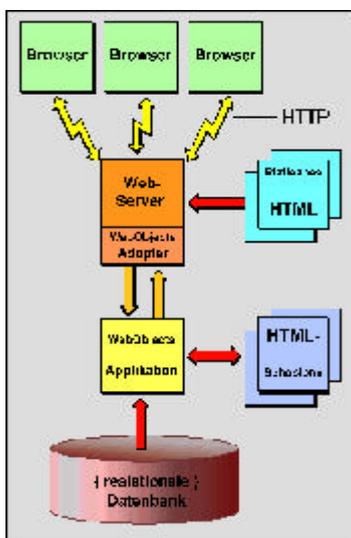
Die erste Version wurde bereits 1994 als Erweiterung des Betriebssystems Nextstep vorgestellt. Seitdem wurde die Software kontinu-

ierlich weiterentwickelt, ohne die Grundlagen wesentlich zu ändern.

Ausgehend von einer HTML-Schablone werden auf der HTML-Seite zusätzliche Objekte plazierte, mit denen auf dem Server Verbindungen zu anderen Datenquellen hergestellt werden können. Die Daten können quasi in jeder beliebigen Form aufgenommen und in die Seite eingepaßt werden, als Datenbank, als einfache Dateien oder als direkt durch den Aufruf der Seite generierte Variable. Im Vordergrund stehen zur Zeit sicherlich die Datenbanken, die als Basis eines Internetshops benötigt werden. Die automatische Generierung von Daten dürfte aber in Zukunft weitere Anwendungsmöglichkeiten eröffnen, denn faktisch behandelt diese Sonderform der Client-Server-Applikationen die HTML-Seiten als nichts anderes als die besondere Form einer grafischen Oberfläche.

Funktionsweise

Die besondere Schwierigkeit beim Umgang mit dynamisch erzeugten HTML-Seiten und dem zugrundeliegenden HTTP-Protokoll ist der Austausch der Informationen. Das HTTP ist ein sogenanntes zustandsloses Protokoll, dem Interaktionen überhaupt nicht entsprechen. Es wurde entwickelt, um formatierte Textdaten anzufordern und zu liefern. Darüber hinaus reichende, komplexere Transaktionen waren nicht vorgesehen. Mittlerweile ist das Protokoll in Grenzen erweitert, an den Grundzügen än-



Erzeugen einer Musterseite als HTML-Schablone, hier als „frame“

dert das jedoch nichts. Vor allem können keine Informationen gespeichert werden, die mehr als den eigentlichen Datenaustausch betreffen.

Es entstehen dadurch zwei Problemkreise. Einerseits müssen Wege gefunden werden, auf denen komplexe Transaktionen über das begrenzte Protokoll ausgetauscht werden können; und es müssen Techniken entwickelt werden, wie die Informationen über das Protokoll zu neuen Transaktionen führen können.

URL

Der erste Schritt ist relativ einfach. Die Information, die der Browser auf der Benutzerseite dem Web-Server abverlangt, ist die Seite, die der Nutzer zu sehen wünscht. Dazu sendet der Browser eine URL, meist in Form eines längeren Dateinamens.

Bei den meisten Applikationsservern, so auch bei WebObjects, wird diese URL dazu benutzt, zusätzliche Informationen zu liefern. Dazu wird sie erweitert, ohne dem Protokoll zu widersprechen. Dem Browser wird vorgespiegelt, daß es sich um die Datei einer einfachen HTML-Seite handele. Auch dem Benutzer fällt, wenn überhaupt, diese URL nur durch ihre reine Länge und durch ihre kryptischen Zeichenfolgen auf, die an die Anfänge der Programmierung erinnern.

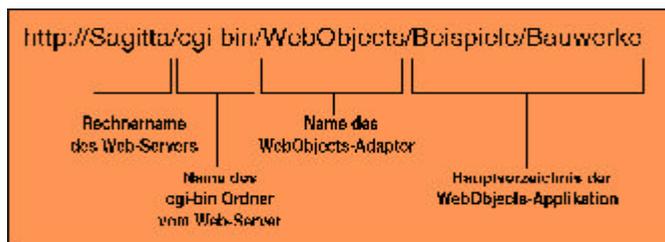
Anforderungen

Die Schwierigkeiten für den Server deuten sich bereits an. Er muß mindestens zwei verschiedene Zustände abspeichern. Der erste ist die Seite, die dem Nutzer gerade zur Verfügung gestellt wird. Auf ihr müssen alle im Programmablauf denkbaren Zustände als URL so realisiert sein, daß sie im Brow-

serfenster als Hyperlinks verfügbar sind. Gleichzeitig muß der Server auf alle URL antworten können. Dafür müssen alle Informationen auf der Seite so lange gespeichert werden, bis der Server eine Antwort erhält. Erst dann kann der Server – theoretisch – die alte Seite verwerfen und die neu angeforderte Seite erzeugen. Leider haben alle Browser Funktionen zum Vor- und Zurückblättern, teilweise auf ihrem lokalen Rechner. Dadurch wird die Sache zusätzlich kompliziert und erfordert besondere Techniken.

Faktisch müssen darum, um den Ablauf einer Sitzung kontrollierbar zu halten, nicht nur die beiden Zustände, sondern alle wichtigen Daten einer Sitzung komplett vorgehalten werden. Das sind zuvorderst die Informationen, welche Seiten abgerufen wurden und welche Hyperlinks bereits erzeugt sind. Genauso aber sollten der Verlauf der Sitzung und zusätzliche Informationen zum Benutzer verfügbar sein. Alle diese Daten müssen auf dem Server abrufbar gespeichert sein.

Darüber hinaus müssen Mittel und Wege gefunden werden, wie über das HTTP nichts weniger als die Ein- und Ausgabe eines Programmes realisiert werden kann.

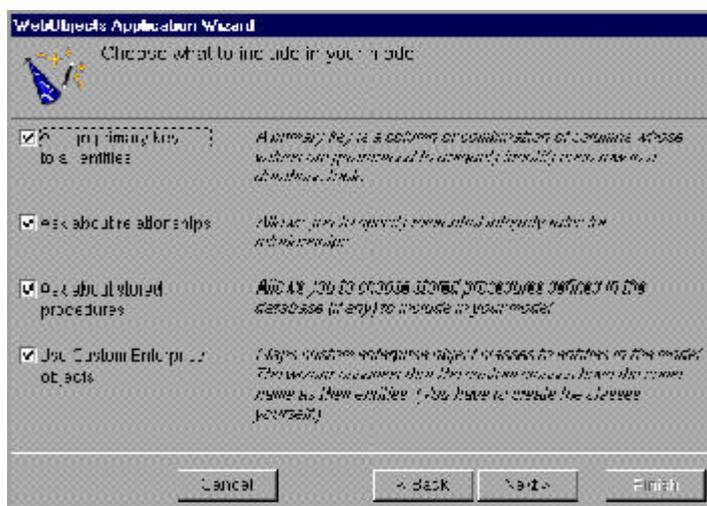


Aufbau der URL, mit der eine dynamisch generierte Seite nachgefragt wird

Bibliotheken

Typischerweise werden komplexe Anforderungen an die Handhabung von Daten in einer von vornherein objektorientierten Umgebung selbstredend mit Objekten realisiert. Mit der Entwicklungsumgebung von WebObjects werden darum umfangreiche Objektbibliotheken (frameworks) zum Erstellen der Applikationen zur Verfügung gestellt. Sie lassen sich grob in vier Teile gliedern:

- Foundation: Basisklassen für Objekte mit allgemeinen Funktionen für Zeichenketten, Datumswerten u. ä.
- AppKit: Bauteile wie Menüs, Fenster, Dialogboxen usw. für die grafische Benutzeroberfläche. Für die Entwicklung von Internet basierten Web-Applikationen sind sie ohne Bedeutung.
- Enterprise Objects (EOF): Komponenten, die einen möglichst effizienten Zugriff auf unter-

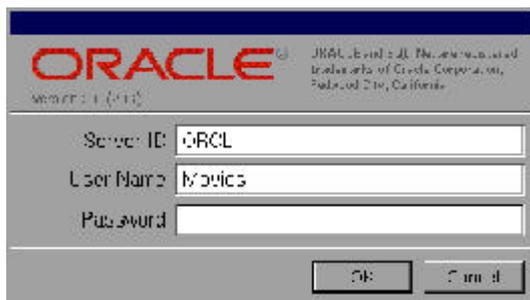


schiedliche Datenquellen ermöglichen. Insbesondere zum Zugriff auf Datenbanken stellt das EOF standardmäßig verschiedene spezifische Datenbankadaptores (u. a. Oracle, Sybase, Informix) sowie einen universell einsetzbaren, objektorientierten Adaptor zur Verfügung.

- **WebObjects (WOF):** Unter der Namensbezeichnung sind die Klassen zusammengefaßt, die zur Entwicklung der eigentlichen Webanwendung notwendig sind. Sie ermöglichen das Ankoppeln an den Server, enthalten die Verwaltungsparameter von Sitzungen und Applikationen und die Funktionen zum Gestalten der dynamischen HTML-Seiten. Einige wichtige Klassen aus dieser Bibliothek werden kurz vorgestellt.

WOApplication

Das zentrale Objekt ist WOApplication, mit dem die einzelnen Sitzungen auf dem Server verwaltet und die Verbindungen zu den anderen notwendigen Programmen



auf dem Server realisiert werden. Das Objekt ist innerhalb der Applikation global verfügbar und kann von jedem anderen Objekt aufgerufen werden.

WOSession

Sobald ein Nutzer eine Sitzung be-

ginnt, muß von der Applikation ein neues Objekt instanziiert werden, daß die einzelnen Seiten verwaltet und die Verlaufsdaten speichert. Dafür existiert die Objektklasse WOSession.

Eine Instanz wird so lange vorgehalten, bis entweder die Sitzung regulär beendet oder ein vorbestimmtes Zeitintervall überschritten wird.

WOComponent

Jede von WebObjects generierte Webseite besteht aus einem HTML-Gerüst, dem Programmcode und zusätzlichen Informationen zur Verknüpfung zwischen den dynamischen Elementen und den Variablen im Quelltext des Programmes. Zusammengefaßt werden diese Informationen immer in einem besonderen Objekt, das die Instanz einer Unterklasse von WOComponent sein muß. Einzelne spezifische HTML-Seiten sind auf diese Weise immer aus einer oder mehreren verschiedenen WOComponenten zusammengesetzt.

Die WOComponenten bilden die eigentliche Grundlage der programmgesteuerten Generierung.

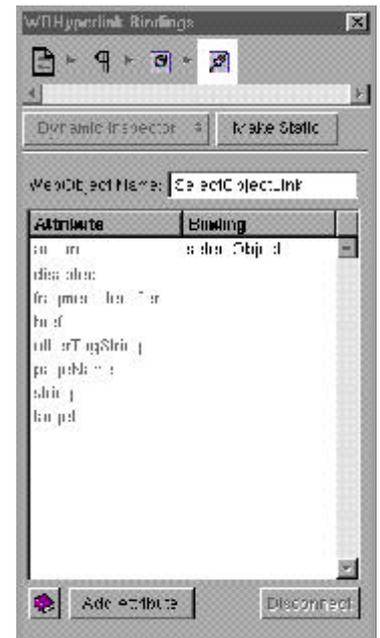
Wird dabei der innerhalb der objektorientierten Programmierung grundsätzliche Gedanke der Wiederverwendbarkeit der Objekte berücksichtigt, ergeben sich weitreichende Möglich-

keiten in der Flexibilität und der Mächtigkeit der Applikationen.

Dynamische Elemente

Es gibt eine ganze Reihe dynamischer Elemente wie WOHyperlink oder WOApplet, die in den HTML-Schablonen eines WOComponent

eingebettet sein müssen. Sie bilden den dynamischen Teil der Komponenten. Bei der Entwicklung der Applikation werden sie deklariert, so daß ihnen als Variable Werte zugewiesen oder entnommen werden können. Erst dann kann die eigentliche Programmierung überhaupt in Angriff genommen werden.



Die einfache Handhabung der Ein- und Ausgabe über Variablen mittels dynamischer Elemente inmitten einer äußerst komplexen Umgebung ist der wesentliche Vorzug der Programmierumgebung von WebObjects.

Programmiersprachen

Die Bibliotheken werden in zwei Programmiersprachen, Java und Objective-C, und einer Skript-Sprache angeboten.

Das war nicht immer so. In den ursprünglichen Versionen existierte als einzige unter Nextstep vorhandene Programmiersprache nur Objective-C. Die etwa zur gleichen Zeit wie WebObjects 1.0 entstandene Programmiersprache Java

steht erst seit WebObjects 3.5 serverseitig komplett bzw. überwiegend auf der Benutzerseite zur Verfügung. Waren die ursprünglichen Klassen noch weitgehend ein reiner Java-Aufsatz auf die Objective-C-Klassen, so sind mittlerweile fast alle Klassen durchgängig in Java und Objective-C ausprogrammiert und die Java-Klassen erreichen das Laufzeitverhalten der Objective-C-Klassen. Trotzdem kann es noch vorkommen, daß innerhalb einer Java-Entwicklung Fehlermeldungen aus der Umgebung von Objective-C auftauchen.

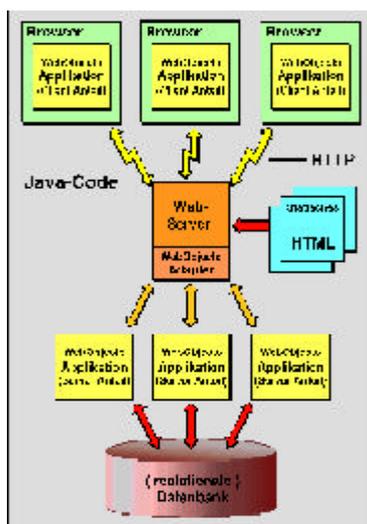
Zum schnellen und einfachen Entwickeln von Applikationen wurde WebScript entwickelt. Diese Sprache benötigt als Skriptsprache keine Kompilierung, sondern nur einen Interpreter.

In der Syntax ist die Sprache sehr an Objective-C angelehnt, bildet aber nur eine Untermenge davon. Sie hat deswegen grundsätzlich einige Vereinfachungen. So existieren überhaupt keine Datentypen. Variablen werden nicht deklariert, sondern zur Laufzeit als Objekte initiiert. Die bekannten Zeiger und Strukturen aus C gibt es ebenfalls nicht mehr. Dafür müssen zusätzliche Methoden gemacht werden, sondern sind zur Laufzeit implizit deklariert.

Im Laufzeitverhalten ergeben sich für Webscript, glaubt man den Entwicklern, keine nennenswerten Einbußen. Die Entwicklungszeit selbst wird aber sehr beschleunigt, da jedes Textfragment direkt getestet werden kann.

Java

Für einige Applikationen ist das HTTP-Protokoll einfach zu restriktiv, als daß vernünftigerweise eine sinnvolle Kommunikation über das Internet etabliert werden sollte. Für solche Fälle bietet WebObjects die Möglichkeit, eigene Java-Clients



innerhalb der dynamischen HTML-Seiten ablaufen zu lassen, also Java-Applets innerhalb des Browserfensters. Zusätzlich zur bestehenden Kommunikation werden Informationen direkt von den Java-Objekten von der WebObject-Applikation über das bestehende Interface entgegengenommen, indem die Java-Objekte der Nutzerseite direkt mit besonderen Objekten auf der Anwenderseite verbunden sind.

Diese Option besteht aber nur für Datenbankanwendungen. Apple hat hier zu den grafischen Elementen, die als

„Swing-Klassen“ in Java zur Benutzerführung entwickelt wurden, besondere Klassen im EOF entwickelt. Auf diese Weise können Eigenschaften des Benutzerrechners in Anspruch genommen werden, so daß dieser einen Anteil der Applikation ausführt. Dadurch werden der Webserver und das Netz entlastet. Eine Applikation unter WebObject kann also durchaus, muß es aber nicht, eine Mischung aus dynamisch generierten HTML-Seiten und einer guten Portion im Browser unter Java ablaufenden Oberfläche sein.

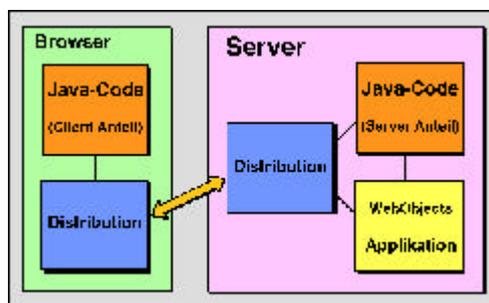
Entwicklerseitig kann dazu der alte Interface-Builder herangezogen werden, mit dem bereits unter Nextstep auf einfache Art die Benutzeroberflächen erzeugt wurden.

Nachteilig ist an dieser Methode nur, daß ein neuerer Browser benötigt wird, der Java und dessen Swing-Klassen beherrscht. Inwieweit auf der nicht kontrollierbaren Nutzerseite Ansprüche an die Leistungsfähigkeit formuliert werden, ist aber ein generelles Problem im Internet. Es bleibt letztlich in der besonderen Verantwortung jedes einzelnen Anbieters, welche Forderungen er stellt oder welche Rücksichten zu nehmen er bereit ist.

Java-Objekte auf der Server- und Nutzerseite

Das Frage-Antwort-Spiel

WebObject-Applikationen werden von Ereignissen der Benutzer gesteuert. Anstelle von Maus- oder Tastaturereignissen werden jedoch

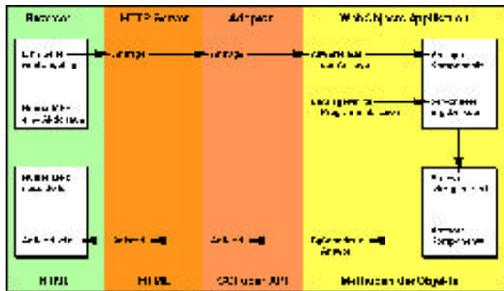


Direkte Kommunikation zwischen einem Java-Objekt im Browser und einem Java-Objekt auf dem Server

Antworten auf eine bestimmte HTML-Anfrage erzeugt. Zu diesem Zweck gibt es, wie bei jeder grafischen Benutzeroberfläche, eine Hauptschleife zum Abfangen der Ereignisse. Sie ist im Falle von WebObjects lediglich komplexer aufgebaut.

Bei jedem Durchlauf der Schleife empfängt die Applikation eine HTTP-Anfrage, antwortet ihr und wartet auf die nächste Anfrage. Dieses Spiel wird solange fortgesetzt, bis es abgebrochen wird, sei es durch regelgerechte Komplettie-

zung der Anfragen, durch überschreiten einer einstellbaren Frist infolge des Abwandern des Besuchers oder durch andere Ereignisse.



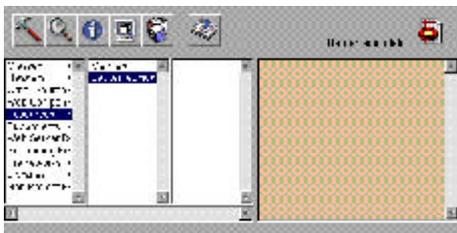
Frage-Antwort-Schleife

Besondere Programme

Eine ganze Reihe von Programmen sind erforderlich, um Applikationen objektorientiert zu entwickeln und mittels dynamisch generierter HTML-Seiten ausführen zu können. Neben Standardapplikationen sind dazu einige spezielle Programme, vor allem als Laufzeitumgebung, erforderlich.

Entwicklungs-umgebung

Natürgemäß von besonderem Interesse sind die Werkzeuge, mit denen Entwickler gezwungener-



Dateiauswahl im ProjectBuilder

maßen ihre Applikationen erstellen müssen. Im Falle von WebObjects ist das der ProjectBuilder, in dem alle notwendigen Werkzeuge, Editor, Compiler, Debugger und

einige andere mehr, integriert sind. Die meisten der Werkzeuge stammen aus der alten Entwicklungsumgebung von Next und sind, mit Ausnahme des WebObjectsBuilder, seit langem bekannt.

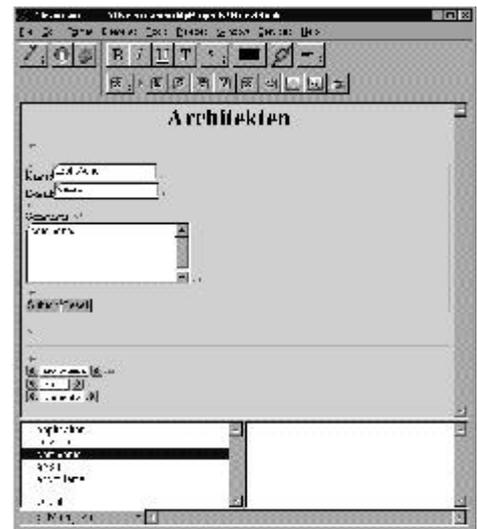
Laufzeit-umgebung

In der Laufzeitumgebung werden zum Bearbeiten der Abfragen und Antworten einige besondere Programme benötigt. Mit ihnen werden die HTML-Seiten dynamisch generiert, und sie steuern letztlich den Ablauf der Sitzungen.

WebObjects – Adaptor

Der Adaptor ist die ausführende Schnittstelle zwischen der eigentlichen WebObjects-Applikation und dem Web-Server. Um das Programm anzusprechen, sind serverseitig in den HTML-Schablonen besondere Marken als sogenannte HTML-Tags eingearbeitet. Sie haben die Form `<Webobject>` und `<\Webobject>` und entsprechen damit in der Syntax den bekannten HTML-Konventionen. Im Vergleich zu anderen Marken werden sie aber nur intern verwendet.

Der Inhalt zwischen den Marken, zumeist Verweise auf dynamische Elemente, wird mittels des Adaptor-Programmes durch die spezifisch ermittelten Werte ersetzt, die sich aus dem Ablauf des



WebObjectsBuilder zum Erzeugen einer HTML-Seite

eingebetteten Programmtextes und seinen Ergebnissen ergeben.

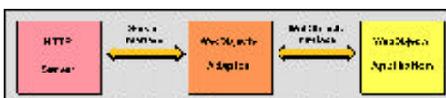
Der Nutzer sieht also diese Marken und den Inhalt dazwischen, sondern nur die vom Adaptor bekanntgegebenen Ergebnisse.

WebObjects – Monitor

Zur Laufzeit ist der WebObject-Monitor das wichtigste Werkzeug. Mit diesem Programm werden die Applikationen konfiguriert und die notwendigen Instanzen auf einem oder mehreren Servern gebildet, wobei auch entfernte Rechner angesprochen werden können. Zu diesem Zweck bietet das Programm eine einfache grafische Benutzeroberfläche, um Applikationen zu starten, Server an- oder abzumelden und dergleichen mehr.

Beispielsweise ist es möglich, bei mehreren gleichen Applikationen in bestimmten Zeitintervallen einen Prozeß herunterzufahren und an dessen Stelle einen neuen zu starten, ohne daß es bemerkt wird.

Ähnliches gilt für fehlerbehaftete Prozesse, die selbstständig neu gestartet werden müssen, wobei die automatische Meldung an



den Administrator nicht fehlen darf. Zusätzlich gibt es umfangreiche Statistikfunktionen und ein Aufnahme- und Abspielprogramm

Implizit wird damit schon deutlich, daß es für den Webserver keine besondere zusätzliche Unterstützung bedarf und prinzipiell jede

Hardware unterstützt werden kann. Selbstverständlich kann aber, um Konflikte zu vermeiden, immer nur jeweils eine Schnittstelle zu einem der unterschiedlichen Webserver bedient werden

Plattformen

Die Laufzeitumgebung von WebObjects ist unter vier Betriebssystemen

lauffähig, unter HP-UX, MacOS X Server, NT und Solaris, die Entwicklungsumgebung allerdings nur für die zwei Plattformen MacOS X Server und Windows-NT 4.0. Inwieweit Apple als Hardwarehersteller das dahinterstehende und von Next übernommene Multiplattformkonzept weiter verfolgt, also auch zukünftig die Hardware anderer Hersteller unterstützt, kann derzeit nicht beurteilt werden.

WebObjects unter NT

Es gibt für Windows-NT 4.0, da es als einziges der unterstützten Betriebssysteme nicht auf Unix aufbaut, eine besondere Variante des Softwarepaketes.

Bei seiner Installation wird zunächst eine abgespeckte Form von OpenStep Enterprise, eine spätere Fassung des Nextstep-Betriebssystem mit Unix-Kern, aufgespielt. Mit der dazugehörigen Terminal-Applikation ist auf diese Weise ein fast kompletter Satz Unix-Befehle im System implementiert. Hinzu kommen mit der etwa 400 MB umfassenden Software einige andere nützliche Unix-Werkzeuge, aber beispielsweise auch, ganz Next-spezifisch, ein Display-Post-

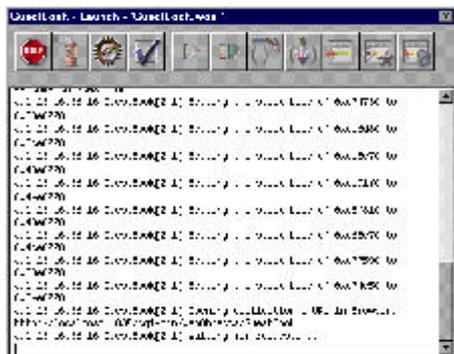
Script-Interpreter, mit auf den Rechner. Wesentlich ist für WebObjects jedoch die besondere Entwicklungsumgebung. Sie ist auf Unix aufgebaut, darum muß es selbst unter NT vorhanden sein. Als Nebeneffekt dieser radikalen Lösung unterscheidet sich die NT-Variante nur in Kleinigkeiten von der OS-X-Version.

Ein Schnäppchen?

Beim Einkauf von Next ist Apple das Softwarepaket WebObjects quasi in den Schoß gefallen. Tatsächlich wird unter diesem Namen, auch unter MacOS X, fast die ganze objektorientierte Entwicklungsumgebung verkauft, die Next seinerzeit für seine eigenen Maschinen entwickelt hat. Dazu zählen die Programmiersprache Objective-C, der „ProjectBuilder“ und der „InterfaceBuilder“ und die Compiler. Auch ohne Ambitionen im Internet lassen sich daher mit WebObjects für das Betriebssystem „MacOS X Server“ und seiner spezifischen grafischen Benutzeroberfläche vollwertige Programme erstellen. Vor ein paar Jahren waren das noch gewichtige Argumente. Seitdem aber Java populär geworden ist und spitze Zungen behaupten, für Software solle kein Geld mehr ausgegeben werden, haben sich die Meinungen verschoben. Trotzdem stellt der Hardwarehersteller Apple sein eigenes Licht eindeutig unter den Scheffel.

Anwendung

Derzeit wird WebObjects vor allem nachgefragt, um in möglichst kürzester Zeit Internet-Shops mit Katalogen aus Datenbanken, Warenkorbfunktionen und dem anderen Zubehör zu erzeugen. Die Entwickler, die in diesen Bereichen tätig sind, erwarten dabei von ihren Kunden eine gewisse Größe. Nur



Ausgabe des Monitors während eines Probelaufes

(recording) für einzelne Sitzungen. Wie alle Programme unter Unix kann dieses Werkzeug mit einem Befehl aus einer Kommandozeile gestartet werden. Hilfreich ist das bei entfernten Prozessen und bei den bekannten Unix-Skripten.

Server-Schnittstellen

Mit dem Paket werden zur Verbindung mit einem Webserver zwei verschiedene Formen von Schnittstellen angeboten. Zum einen wird als generelle Variante die cgi-Schnittstelle bedient, die sich als Quasi-Standard durchgesetzt hat und mit der so gut wie alle gängigen Webserver ausgestattet sind.

Darüber hinaus gibt es einige spezielle API für besonders populäre Webserver, nämlich ISAPI, das unter Microsoft-Windows-NT-4.0-Server für den IIS bzw. „Peer Web“ in der Workstation-Version von NT zur Verfügung steht, NSAPI für den Webserver von Netscape 3.5.1 (NT und OS X) und, ausschließlich in der MacOS-X-Variante, eine weitere für den Webserver von Apache. Letzterer wird derzeit mit OS X und WebObjects gebündelt von Apple auf den Server-Maschinen ausgeliefert.

solche sind derzeit in der Lage, bei der unvorhersehbaren Nachfrage im Internet schnell einen oder mehrere großzügig dimensionierte Server bereitzustellen, um die notwendige Performanz zu bieten. Bei diesen Applikationen, für die im Bauwesen wenig Verwendung besteht, liegt eindeutig ein Schwerpunkt.

Kosten

Darüber hinaus ist die Preisgestaltung sehr offensiv. Zwar ist die Einstiegsvariante unter MacOS X kostenfrei, aber die Nutzung ist hierbei auf 50 Transaktionen in der Minute auf nur einem Server begrenzt.

Ohne Maschine und Betriebssystem muß für das gleichartige Angebot unter NT etwa das Gleiche bezahlt werden.

Unbegrenzte Transaktionen mit nur einem Server kosten bereits deutlich fünfstelligen Beträge, die Highendversion mit zusätzlich unbeschränkter Anzahl von Servern sogar fast 100 000 DM. Damit sind noch nicht alle Kosten gedeckt, denn selbstverständlich muß, meist als Unikat, noch eine Applikation entwickelt werden.

Bauwesen

Die Werkzeuge, wie sie, nicht nur von WebObjects, mittlerweile mit der Entwicklung des Internet zur Verfügung stehen, bieten bei der besonderen Arbeitsform im Baubereich, der Produktion von Unikaten mit wechselnden Beteiligten, eine interessante und überdenkenswerte Alternative.

Insbesondere im Bereich AVA ergeben sich eine Reihe von po-

tentiellen Möglichkeiten zum Steigern der Effizienz und der Produktivität.

Dennoch gibt es im Bauwesen derzeit keine bekannten Applikationen, unter denen WebObjects eingesetzt wird, auch nicht im Bereich der Hochschulen. Dazu ist das Produkt, obwohl sehr leistungsfähig, zu unbekannt.

Resümee

WebObjects bietet, nicht zuletzt dank seiner Herkunft als Erweiterung eines der ersten vollständig objektorientierten Betriebssysteme, einen ungewöhnlichen, aber einfachen Weg, um die modernen Techniken des Internet schnell und angemessen einsetzen zu können. Der insgesamt positive Eindruck wird allerdings durch den hohen Preis getrübt.

Glossar:

API: Application Protocol Interface

cgi: common graphic interface

HTML: Hypertext Markup Language

HTTP: Hypertext Transfer Protocol

URL: Union Request Link

Web-Site: Internetadresse, unter der HTML-Seiten über das HTTP angefordert werden können.

Literaturhinweise:

Apple (Hrsg.): WebObjects Developer Documentation [<http://www.developer.apple.com/techpubs/webobjects/>]

Lipps, Peter: WebObjects – Objekte im „World Wide Web“, in: Objektspektrum 2/96.

Lonsing, Werner: Objektorientierte Programmierung, in: bauinformatik, Nr. 6/93.

Zepf, Matthias: Zufallsfund, in: iX, Nr. 6/99, S. 50 ff.

Eine kleine Auswahl an Webseiten, die WebObjects einsetzen und die öffentlich zugänglich sind:

- www.libri.de
- www.tui.de
- www.cornelsen.de
- www.matsushita.de
- www.csd.toshiba.com/cgi-bin/WebObjects/Toshiba
- www.store.sendamac.de/index.html
- www.geschenkemarkt.de
- www.packardbell-europe.com
- www.apple.com/germanstore
- www.store.disney.go.com
- www.hardrock.com
- www.application.mercedes-benz.com/WebObjects/BIGOnlineD
- www.v2music.co.uk
- www.bosch-stuttgart.intershop.de/scripts/WebObjects-ISAPI.dll/BoschUCL

► Onlinewerbung und -planung:

- www.plan-net.de/media-system

► Banken:

- www.deutsche-bank.de
- www.fcmbd.com/cgi-bin/WebObjects/rates/mortgages
- www.meine.bank24.de

► Stadtinformationen:

- www.berlin.de

► Nachrichten:

- www.bbc.co.uk/home/today