Datenbankanbindung an das WWW
- Techniken, Tools und Trends -

Henrik Loeser
Universität Kaiserslautern, FB Informatik,
Postfach 3049, 67653 Kaiserslautern
email: loeser@informatik.uni-kl.de

Zusammenfassung: Das rasante Wachstum des World Wide Web (WWW) hat eine allgemein akzeptierte, plattformunabhängige Basis für die Informationsbereitstellung und -präsentation geschaffen. Um große Informationsmengen einfach zur Verfügung stellen und Geschäftsvorgänge sicher verwalten zu können, bedarf es des Einsatzes von Datenbanksystemen (DBS), was die Anbindung von DBS an das WWW impliziert. An die DBS ergeben sich, wenn man WWW-basierte Anwendungen unterstützen möchte, neue Anforderungen. Sie entstehen u. a. durch die neuen multimedialen Datentypen, die stark wachsenden Informationsmengen, den weltweiten Zugriff sowie die Koexistenz zu Legacy-Applikationen. In diesem Beitrag stellen wir zunächst die derzeit vorhandenen Techniken zur DB-Anbindung vor und zeigen künftige Entwicklungen. Wir klassifizieren die unterschiedlichen Techniken, analysieren ihre Stärken und Schwächen und geben Beispiele. Daran anschließend erarbeiten wir die sich an DBS ergebenden neuen Anforderungen.

1. Einleitung

Das World Wide Web (WWW) und die dahinter stehende Technik erleben seit 1994 einen rasanten Zuwachs. Täglich kommen neue WWW-Server hinzu, Unternehmen begeben sich in das Internet oder bauen Intranets zum internen Informationsaustausch auf. Der elektronische Handel (Electronic Commerce) beginnt sich zu entwickeln, da immer mehr potentielle Kunden erreichbar sind. Durch das Interesse der Medien am WWW weiß mittlerweile ein Großteil der Bevölkerung etwas mit dem Stichwort „Datenautobahn" anzufangen. Über sie wurden in den Anfangstagen des WWW nur statischer Text und Bilder in Form von HTML-Seiten transportiert. Datenbankzugriffe über das Internet wurden wie bisher über Gopher- und andere Schnittstellen abgewickelt. Seit dem Betreten der Präsentations- und Kommunikationsplattform WWW durch zahlreiche Firmen und Institutionen ist ein zunehmender Bedarf an aktuellen und häufig aktualisierten Informationen festzustellen. Zudem müssen für den elektronischen Handel Geschäftsvorgänge im WWW abgewickelt und Kundendaten bereitgestellt werden. Für die Datenverwaltung und -bereitstellung kommen nur Datenbanksysteme (DBS) in Betracht, was die Anbindung von Datenbanken an das WWW impliziert.

Dieser Beitrag gliedert sich wie folgt: Im nun folgenden Abschnitt stellen wir kurz die Geschichte und wesentliche Konzepte des WWW sowie die Programmiersprache Java vor. Anschließend motivieren wir anhand von Beispielen die unterschiedlichen Einsatzmöglichkeiten für Datenbanksysteme in Bezug auf das WWW. Der darauf folgende Abschnitt stellt die bisherigen Evolutionsstufen von Datenbankanbindungen an das WWW vor, gibt Beispiele, analysiert ihre Vor- und Nachteile und gibt einen Ausblick auf kommende Entwicklungen. Im daran anschließenden Abschnitt erarbeiten wir Anforderungen an kommende Datenbanksysteme bezüglich der Unterstützung des WWW und zeigen die daraus resultierenden Probleme auf. Den Abschluß dieses Beitrags bildet eine Zusammenfassung der hier vorgestellten Ergebnisse in Verbindung mit einem Ausblick auf zukünftige Arbeiten.

2. Grundlagen

Das World Wide Web ist dabei, Einfluß auf viele Bereiche unseres Lebens zu nehmen, die Programmiersprache Java könnte es dabei unterstützen. In den beiden folgenden Abschnitten stellen wir zunächst das World Wide Web, dann Java vor.

2.1 Das World Wide Web

Die Geschichte des World Wide Web (WWW) geht auf einen Vorschlag von Tim Berners-Lee vom Europäischen Forschungszentrum für Teilchenphysik (CERN) in Genf aus dem Jahre 1989 zurück, als dieser seinen Entwurf für ein Hypertext-System zum ersten Mal vorstellte [Cal96]. Seit dem Beginn der ersten Arbeiten Ende 1990 sind mittlerweile zahlreiche Browser (WWW-Clients) erschienen, d. h. Programme zur Darstellung der mit HTML (HyperText Markup Language) erstellten Hypertext-Dokumente. Die Entwicklung von HTML, deren Sprachnorm vom mittlerweile entstandenen WWW-Konsortium verwaltet wird, wurde vor allem durch anbieterspezifische Erweiterungen, die später in den Standard aufgenommen wurden, vorangebracht.

Die Hypertext-Dokumente werden auf Anforderung des Browsers mit Hilfe des zustandslosen HTTP (HyperText Transfer Protocol) von einem WWW-Server (HTTP-Dämon) zum Client, dem Browser, übertragen (Abb. 1, (1) und (2) in Abb. 3). Für jede Übertragung muß eine separate Verbindung etabliert werden. Das in den Browser zu ladende Dokument wird durch eine URL (Uniform Resource Locator, z. B. “http://www.uni-kl.de/AG-Haerder/AG-HaerderD.html”) spezifiziert. Sie enthält die zu benutzende Protokollart, den Server-Namen sowie weitere Angaben, meist Verzeichnis- und Dateinamen. Neben HTTP können auch andere Internet-Protokolle (FTP, Telnet, News, ...) angegeben werden.

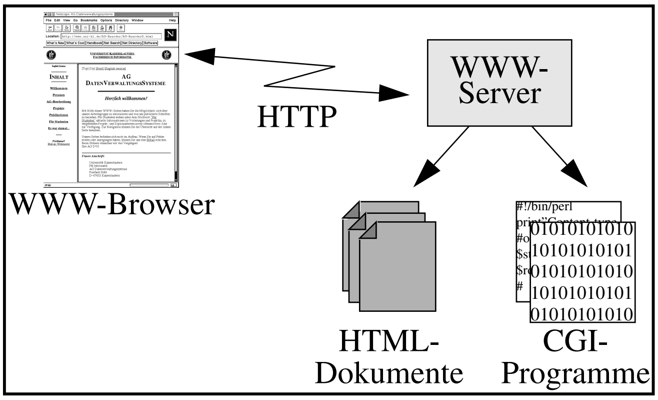

Abb. 1: WWW-Mechanismus

Um nicht nur fest spezifizierte Dateien oder solche mit statischem Inhalt einladen zu können, wurde das Common Gateway Interface (CGI, [CGI95]) entwickelt, mit dessen Hilfe auf dem Server bereitgestellte Programme dort ausgeführt und deren Ergebnisausgabe nach der Terminierung anstelle einer Datei zum Client übertragen wird. Hierzu muß dem Server statt eines Dateinamens der Name des ausführbaren Programmes zusammen mit optionalen Kommandozeilenparametern übergeben werden. Diese Parameter stammen, wenn sie nicht fest vorgegeben wurden, meist von HTML-Formularen. Sie gestatten dem Anwender die Eingabe von Text, die Auswahl aus verschiedenen Optionen usw. innerhalb des im Browser angezeigten Dokuments. Drückt der Benutzer nun nach der Eingabe der Daten auf den Ausführungsknopf (Submit-Button), so werden die Daten als Parameter an ein auf dem Server liegendes Programm geschickt. Dieses kann dann in Abhängigkeit von den übertragenen Informationen antworten und z. B. das Ergebnis einer Datenbankanfrage übertragen.

Um Daten sicher zwischen Browser und Server übertragen zu können, stehen zur Zeit zwei verschiedene Verfahren zur Verfügung. Dies ist zum einen SHTTP (Secure HTTP, [RS95]), eine Erweiterung des HTTP um Verschlüsselungstechnik, zum anderen SSL (Secure Socket Layer, [FKK96]), ein von Netscape entwickeltes Übertragungsverfahren, das die Verschlüsselung auf der Socket-Ebene einführt und somit auch von anderen Protokollen genutzt werden kann. Hierzu wurde u. a. die neue Protokollart HTTPS, HTTP mit SSL, eingeführt. Da SSL bereits in zahlreiche Browser und Server integriert ist und gegenüber SHTTP Vorteile besitzt, wird es wohl bald als Standard akzeptiert werden und SHTTP völlig verdrängen.

2.2 Java

Wurde HTML und damit das WWW bislang lediglich um weitere Darstellungs- und Eingabemöglichkeiten (Tabellen, Formulare usw.) ergänzt, so bekam es 1995 einen besonderen Impuls durch die Vorstellung der von SUN Microsystems entwickelten objektorientierten Programmiersprache Java [GJS96]. Durch die Interpretation des aus Java-Quelltext für eine virtuelle Maschine erzeugten Byte-Codes wird Plattformunabhängigkeit erreicht. Hierdurch ist es möglich, ein Programm („Applet“) von einem Server in einen diese virtuelle Maschine zur Verfügung stellenden WWW-Browser zu laden und dort lokal auszuführen. Programme können wie Bilder oder Text in Hypertext-Dokumente integriert werden und erlauben so Animationen, Benutzerinteraktionen, Berechnungen sowie vom HTTP unabhängige Netz- und damit auch DB-Verbindungen.

Für Java wurden bislang zahlreiche Erweiterungen spezifiziert, u. a. auch die Java Enterprise API (Application Programming Interface). Sie enthält die Teilpakete JDBC (Java DataBase Connectivity, [HC96]), Java IDL (Interface Definition Language) sowie „Java RMI and Object Serialization“. JDBC stellt ähnlich wie ODBC (Open Database Connectivity) eine Aufrufschnittstelle für SQL zur Verfügung. Die Java IDL spezifiziert die Abbildung der in IDL beschriebenen Objekttypen nach Java. IDL ist eine von der OMG (Object Management Group) im Rahmen von CORBA (Common Object Request Broker Architecture) entwickelte sprachneutrale Schnittstellenbeschreibungssprache. Die dritte Komponente, Java RMI (Remote Method Invocation), stellt einen dem RPC (Remote Procedure Call) ähnlichen Mechanismus zum Methodenaufruf für entfernt gehaltene Objekte zur Verfügung. Um Objekte verlustfrei übertragen zu können, steht ferner Funktionalität zum Packen/Entpacken (Marshalling, Serialization) von komplexen Objekten bereit.

3. DB-Anwendungen und das WWW

Täglich nimmt die Zahl der Angebote im WWW zu. U. a. auch ausgelöst durch den Konkurrenzdruck beginnen immer mehr Unternehmen mit Internet-Aktivitäten. In der ersten Phase steht dabei oft nur die Präsenz, d. h. die Nutzung des WWW als vergleichsweise preiswertes Werbemedium und Marketinginstrument, im Vordergrund. Neben der Bereitstellung von Informationen und anderen kostenlosen Service-Angeboten ist die Feedback-Möglichkeit der Nutzer („Kundendialog“) ein wichtiger Aspekt der WWW-Tätigkeiten, da so auch Informationen über potentielle Kunden gewonnen werden können. In einer zweiten, teilweise gerade erst beginnenden Phase wird das WWW als Geschäftsplattform und zur Optimierung (inner-) betrieblicher Abläufe genutzt. In beiden Phasen und für alle der angesprochenen Nutzungszwecke werden immer häufiger Datenbanksysteme eingesetzt.

Wird den Nutzern eine große Informationsmenge, in der Regel durch sehr viele HTML-Seiten, zur Verfügung gestellt, so bietet sich die Verwaltung der Hypertext-Dokumente mit Hilfe eines DBS an. Einige DBS-Hersteller [Ill95, O295, Odi96a, SAG96] bieten bereits entsprechende Zusatzmodule zur Dokumentenverwaltung, d. h. der Speicherung von HTML-Texten innerhalb einer DB, an. Je nach Produkt sind dabei u. a. Versionierung, Datenkompression, vereinfachte Zugriffsverwaltung sowie (verbesserte) Suchmöglichkeiten zu nennen. Um die gespeicherten Dokumente im Browser anzeigen zu können, bedarf es einer geeigneten Schnittstelle zwischen DBS und WWW-Server.

Die zweite, weitaus häufiger genutzte Einsatzmöglichkeit von DBS im WWW ist die der Verwaltung „traditioneller“ Daten. Diese werden über entsprechende Schnittstellen, wie wir sie im nächsten Kapitel vorstellen und klassifizieren werden, zu einem Browser übertragen oder aus diesem heraus aktualisiert. Im folgenden wollen wir beispielhaft einige Applikationen vorstellen, um den Bedarf an geeigneten Web-Anbindungen von Datenbanken zu motivieren:

- ***Bereitstellung aktueller (Börsen-)Daten***: In Verbindung mit dem Trend zu Direktbanken mit „Online-Banking“ bzw. „Online-Brokerage“ (Bank24/Deutsche Bank, comdirect, Dresdner Bank, Advance Bank, Direkt Anlage Bank, ConSors, Gries & Heissel, etc.) und dem Wegfall von persönlicher Beratung entsteht bei den Kunden ein Informationsbedarf, u. a. nach aktuellen Börseninformationen. Um diesen zu befriedigen, werden nun dem Kunden aktuelle Börsenkurse im WWW zur Verfügung gestellt. Als zusätzlicher Service werden zudem grafisch aufbereitete Kursanalysen (Charts) und kundenspezifische Portfolio-Analysen angeboten. Zudem existieren mittlerweile von verschiedenen kommerziellen Anbietern (Hoppenstedt, Hornblower, etc.) entsprechende, weitergehende Angebote.
- ***Nachrichten(-Ticker)***: Von Zeitungen werden im WWW Ausschnitte der aktuellen Ausgabe sowie ein elektronisches Archiv mit Suchmöglichkeiten angeboten. Da die Zeitungen über einen elektronischen Nachrichtenticker durch die Presseagenturen beliefert werden, ist es ohne personellen Aufwand möglich, dem Benutzer einen Teil der aktuellen Meldungen auf WWW-Seiten zur Verfügung zu stellen. Hierzu ist es notwendig, eingehende Meldungen zu speichern sowie eine Überblicksseite automatisch zu aktualisieren.
- ***Verfolgung von Vorgängen*** (Auftragsbearbeitung, Paketdienst): Setzt ein Unternehmen intern eine elektronische Auftragsbearbeitung oder ein ähnliches, auf die jeweilige Dienstleistung (z. B. Paketdienst) abgestimmtes System ein, so kann über das WWW als Kundenservice ein Zugriff auf den aktuellen Bearbeitungszustand angeboten werden (z. B. UPS, FedEx). Neben dem Prestige-Gewinn durch entsprechende Serviceleistungen können aber auch ökonomische Vorteile in Bezug auf Einsparung von Personalkosten (telefonische Auskunft) erwartet werden.
- ***Bestell-Center***: Viele Unternehmen verwenden das WWW als Plattform für die Firmenpräsentation, verbunden mit dem Angebot einer Produktauswahl und der Möglichkeit zur Informationsanforderung. Hierbei sind die jeweils vom Anwender anzugebenden Daten (Adresse, Kundennummer, etc.) nach einer eventuellen Prüfung in die interne Kunden- und Auftragsdatenbank aufzunehmen.

Neben diesen Anwendungen zeichnen sich zum einen durch den technischen Fortschritt, zum anderen durch die zunehmende Akzeptanz auf Seiten der Benutzer sowie innerhalb der Unternehmen weitere WWW-basierte Einsatzgebiete ab:

- ***Intranet (EIS, WFMS)***: Durch die Einführung von Intranets können auf Basis der Internet- und WWW-Technik unternehmensinterne Anwendungen (plattformunabhängig) realisiert werden. Hierzu gehört zum einen die eines Enterprise-Information-Systems (EIS), zum anderen aber auch Gruppenarbeitssysteme wie z. B. Workflow-Management-Systeme (WFMS). Mit Hilfe eines EIS kann nun allen Mitarbeitern, evtl. mit Abstufungen der Zugriffsrechte, ein Großteil der bisher papierbasierten Informationen wie Nachschlagewerke (Telefonbücher, Anschriftenlisten etc.) und interne Mitteilungen mit Suchmöglichkeiten zur Verfügung gestellt werden. Durch WFMS können (betriebsinterne) Vorgänge gesteuert und koordiniert werden, wobei eine optimale und fehlertolerante Ausführung der Abläufe angestrebt wird. Durch die Plattformunabhängigkeit von HTML und Java ist die WWW-Technik prädestiniert, um als Basis für die Realisierung solcher Systeme in einem meist heterogenen und verteilten Umfeld zu dienen.
- ***Electronic Banking / Electronic Commerce***: Wird das Vertrauen bei den Geschäftspartnern, d. h. Kunden und Unternehmen, in die Sicherheit der eingesetzten Technik weiter gestärkt, so wird der elektronische Handel sowie die Abwicklung von Bankgeschäften drastisch zunehmen. Neben der Einführung von elektronischem Geld ist besonders die Unter-

stützung von Kreditkarten interessant. Die großen Kartenunternehmen arbeiten bereits zusammen mit Industriepartnern an einem entsprechenden Abrechnungsstandard [SET96]. Bislang wird ein Großteil der mit der Hilfe des WWW getätigten Bestellungen intern noch ausgedruckt und wie eine schriftliche Bestellung behandelt. Setzt man jedoch ein WFMS mit WWW-Anbindung ein, so kann die Bestellung durch den Kunden der erste Schritt in einem durch ein WFMS gesteuerten Bearbeitungsablauf sein.

Die dargestellten Beispiele zeigen nur einen Auschnitt der Einsatzmöglichkeiten von DBS im Zusammenspiel mit dem WWW. Das jeweils zur Verfügung stehende Potential wird oft nicht ausgeschöpft, vielfach das WWW überhaupt nicht genutzt. In Zukunft wird die Zahl der WWW-gestützten Anwendungen jedoch stark zunehmen, damit auch die Zahl der angebundenen DBS.

4. Datenbankanbindung an das WWW

Seit der Einführung des Common Gateway Interface (CGI, [CGI95]) ist es durch das Server-seitige Ausführen von Programmen möglich, in Datenbanken gespeicherte Informationen in HTML-Seiten einzubinden. Bei der dabei verwendeten Technik kann man trotz der kurzen Entwicklungsgeschichte schon deutliche Evolutionsstufen erkennen. Zudem ermöglichen derzeitige Trends eine Prognose bezüglich weiterer Entwicklungen. Wir wollen nun die einzelnen Stufen vorstellen und sie auf ihre Vor- und Nachteile hin untersuchen. Dazu betrachten wir Kriterien wie mögliche Transaktionsdauer sowie den Erstellungs- und Wartungsaufwand der zugehörigen HTML-Seiten und Programme.

```
#!/usr/bin/perl
# Msql-Package laden:
use Msql;

# Seitenkopf ausgeben:
print”Content-type: text/html\n\n”;
# [...]

# Verbindung mit dem DB-Server herstellen:
$testdb = Msql->connect;

# Datenbank auswaehlen:
$testdb->selectdb(“bookmarks”);

# Datenbankanfrage stellen:
$sth = $testdb->query(“select name,url from
   bookmarks where url<>’’ order by name”);

# Resultat ausgeben:
print”<TABLE BORDER=1>\n”;
print”<TR>\n<TH>Name<TH>URL</TR>”;
$rows = $sth->numrows;
while ($rows>0)
{
   @sqlrow = $sth->fetchrow;
   print”<tr><td>”,@sqlrow[0],”</TD><td><A HREF=\””,
       @sqlrow[1],”\”>”,@sqlrow[1],”</A></td></TR>\n”;
   $rows--;
}
print”</TABLE>\n”;

# Seitenende ausgeben
# [...]
```

Abb. 2: Web-Anbindung der 1. Generation (mSQL)

4.1 Die erste Generation

Als Datenbankanbindungen der ersten Generation bezeichnen wir die vom Informationsanbieter für die einzelnen DB-Anfragen selbst erstellten Skripte oder Programme (meist Perl, C oder Shell-Skripte). Sie werden durch den CGI-Mechanismus des WWW-Servers aufgerufen und mit entsprechenden Parametern versorgt. Im Skript bzw. dem Programm sind die auszuführende DB-Anfrage und die zu erstellende HTML-Ergebnisseite, in die das Anfrageergebnis eingebunden werden soll, fest kodiert. Die vom DBS gelieferte Antwort, deren Struktur schon bei der Skript- bzw. Programmerstellung weitestgehend festgelegt werden muß, wird gelesen und dann in HTML umgewandelt. Da die als Antwort zu übertragende HTML-Seite und auch die DB-Anfrage fest kodiert sind, kann ein so erstelltes CGI-Programm nur einen bestimmten Auftrag ausführen, d. h., für jede Aufgabe muß ein neues Programm erstellt werden. Abb. 2 zeigt ein einfaches, in Perl geschriebenes CGI-Programm, das aus einer mSQL-DB mit Bookmarks alle gespeicherten Namen von WWW-Seiten zusam-

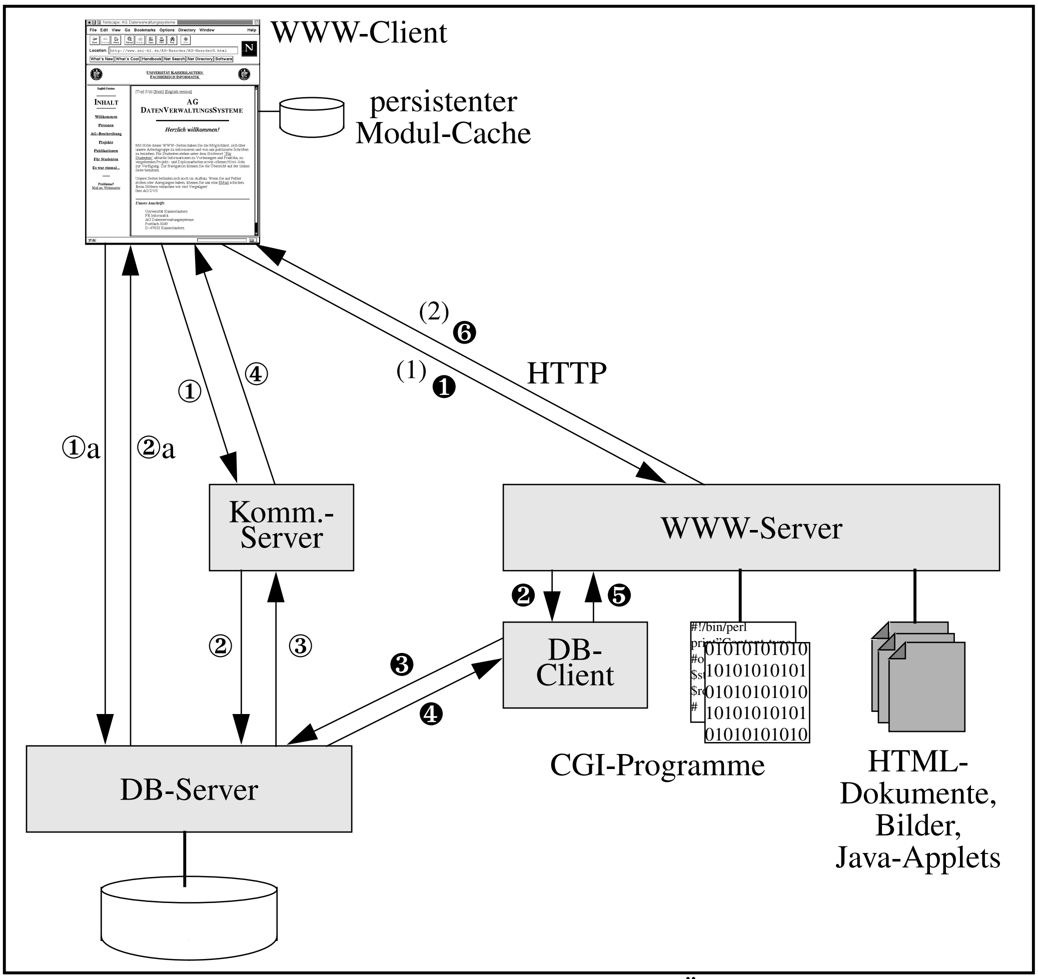

Abb. 3: DB-Anbindung an das WWW (Überblick)

men mit den zugehörigen URLs liest und daraus eine Tabelle innerhalb eines HTML-Dokumentes erzeugt.

Datenbankanbindungen an das WWW mit Hilfe von CGI-Programmen haben mehrere charakteristische Nachteile, die auch die meisten Produkte der zweiten Evolutionsstufe nicht überwinden können. Wir wollen daher diese prinzipiellen Nachteile zuerst, getrennt von den nur dieser ersten Generation eigenen, vorstellen.

- ***Zustandslosigkeit des HTTP***: Das für die Kommunikation zwischen dem WWW-Browser und -Server verwendete HTTP ist zustandslos, was Auswirkungen auf die potentielle Dauer von Transaktionen hat. Da die Verbindung zwischen beiden kommunizierenden Komponenten nur auf den Aufruf des CGI-Programms und die anschließende Ausgabeübertragung beschränkt ist, ist die Dauer von Transaktionen auch nur auf die kurze Abarbeitung des CGI-Programms begrenzt.
  Ein Ausweg, um dieses Problem zu umgehen, besteht z. Z. in der Übertragung von Zusatzinformationen, die im HTML-Dokument als versteckte Variablen (HIDDEN VALUE) eingefügt werden und den aktuellen Bearbeitungszustand repräsentieren. Allerdings können keine Sperren gehalten werden, so daß nur unkritische Operationen möglich sind.
- ***Mehrstufige Aufrufhierarchie***: Für jede Ausführung von DB-Operationen muß eine mehrstufige Hierarchie durchlaufen werden (siehe Abb. 3, ❶-❻). Vom WWW-Server muß jedesmal erneut das gewünschte CGI-Programm gestartet (❷) und dann von diesem zunächst der Verbindungsaufbau zum DB-Server vollzogen werden (❸). Hierdurch ergibt sich ein erheblicher Mehraufwand und eine geringe Ausführungsgeschwindigkeit. Da alle DB-Aufrufe und zusätzlich noch die „normalen“ Dokumentanforderungen über den WWW-Server abgewickelt werden, kann dieser schnell zum Flaschenhals werden.

Hier bieten sich die Installation mehrerer, auf unterschiedliche Rechner verteilter WWW-Server und eine „geschickte" Verknüpfung der Dokumente als Ausweg an.

- ***Unsicherheit über Ausführung***: Der Anwender kann nicht sicher sein, ob eine von ihm intendierte DB-Operation auch ausgeführt wird, da eine Störung der Verbindung vor oder nach der Ausführung des CGI-Programms auftreten kann. Wird der Aufruf vom Anwender wiederholt, so kann es vorkommen, daß eine Operation ein zweites Mal ausgeführt wird.
- ***Mangelnde Zugangskontrolle***: Das CGI-Programm führt, falls nicht über ein HTML-Formular ein Benutzername spezifiziert werden muß, alle DB-Operationen als *ein* DB-Benutzer durch. Es ist also ohne Zusatzmaßnahmen nicht möglich, benutzerabhängig Informationen im Web bereitzustellen.

Neben den allgemeinen Problemen der CGI-Programme besitzen die Web-Anbindungen der ersten Generation noch einige spezifische Nachteile, die von späteren Evolutionsstufen weitestgehend behoben werden:

- ***Schwere Lesbarkeit***: Die in die einzelnen CGI-Programme einkodierten HTML-Seiten sind nicht gut zu erkennen und daher schlecht zu warten. Zudem können die Seiten nicht mit den üblichen HTML-Editoren erstellt werden, sondern sind mühsam als print-Befehle zu realisieren (siehe Abb. 2).
- ***Aufwendige Programmerstellung***: Für jede Aufgabe ist ein neues CGI-Programm zu entwickeln. Werden, wie normalerweise üblich, Benutzereingaben verarbeitet, muß der Programmierer sich um den umständlichen Zugriff auf die vom WWW-Server bereitgestellten Programmparameter selbst kümmern, was Fehleranfälligkeit bedeutet.

Neben diesen vielen Nachteilen besitzen die Web-Anbindungen der ersten Generation auch einen Vorteil, den wir nicht unterschlagen wollen: ***Flexibilität***. Da alle Programme selbst entwickelt werden müssen, steht auch die gesamte Funktionalität des DBS im Rahmen der oben genannten Einschränkungen zur Verfügung.

4.2 Die zweite Generation

Ausgehend von den Problemen mit den DB-Web-Anbindungen der ersten Generation und beschleunigt durch die immense Nachfrage wurden von Datenbankherstellern und Drittfirmen Werkzeuge zum direkten DB-Zugriff mit Hilfe des CGI entwickelt. Wir werden in diesem Abschnitt die wesentlichen Techniken dieser Generation vorstellen und beispielhaft einige Produkte betrachten und klassifizieren.

Techniken

- ***CGI/Server-Erweiterung***: Häufigste Gemeinsamkeit aller Web-Anbindungen der zweiten Generation ist jeweils *ein* allgemeines CGI-Programm [Hug96, NS96, Ill95, O295, Odi96a], das bei jedem DB-Zugriff anstelle der bisherigen Individuallösungen aufgerufen wird. Um die durch den Aufruf eines CGI-Programms entstehenden Geschwindigkeitsverluste zu vermeiden, wird in einigen Ansätzen [Ora96, SAG96, Syb96] eine Erweiterung des WWW-Servers mit Hilfe der entsprechenden Server-API (Application Programming Interface) vorgenommen. Hierdurch findet die Programmabarbeitung im Adreßraum des WWW-Servers statt, ein Aufruf eines CGI-Programms ist nicht mehr erforderlich, und ❷ und ❺ in Abb. 3 entfallen. Ein zusätzlicher Vorteil besteht darin, daß eine ständige Verbindung zum DB-Server gehalten werden kann. Etwaige Wartezeiten zum Verbindungsaufbau und für die Autorisierung fallen nicht mehr an, ❸ wird erheblich beschleunigt.
- ***HTML/Makro/Programm***: Neben den bisher als Programmparametern in der URL übergebenen Anfrageinformationen muß in der Regel nun auch eine Datei spezifiziert werden, deren Inhalt je nach Produkt variiert. Eine häufige Lösung ist der Gebrauch von um Daten-

bankoperationen erweiterten HTML-Seiten. Anfragen, benötigte Variablen etc. werden durch spezielle Befehle (<!msql>, <syb>, <?misql>) in das normale Dokument eingebettet. Ein wesentlicher Vorteil dieser Technik ist die Kontrollmöglichkeit der erstellten Seiten mit Hilfe eines WWW-Browsers. Da die in den Seiten enthaltenen DB-Operationen, d. h. die dem Browser unbekannten Anweisungen, ignoriert werden, wird die Seite in der späteren Form dargestellt. Das in Abb. 4 gezeigte HTML-Dokument, in dem W3-mSQL [Hug96] verwendet wird, hat dieselbe Funktionalität wie das in Abb. 2 dargestellte Perl-Skript, ist jedoch viel einfacher zu erstellen.

Eine andere, zweite Technik besteht in der Auswertung einer Makrodatei, in der getrennt voneinander DB-Operationen und entsprechende HTML-Seiten abgelegt werden. Die Inhalte der unterschiedlichen Sektionen können zwar unabhängig voneinander mit entsprechenden Werkzeugen entwickelt werden, allerdings ist nach dem Zusammenfügen eine spätere Nachbearbeitung nur schwer möglich.

Zum Schluß besteht eine dritte Technik in der Erweiterung der DB-Programmierschnittstelle um Web-spezifische Funktionalität, d. h. der Möglichkeit zur einfachen Generierung von HTML-Ausgaben. Durch die Verwendung der Programmierschnittstelle steht, ähnlich wie bei den Ansätzen der ersten Generation, der volle Funktionsumfang des DBS zur Verfügung, man ist bei der Programmerstellung auf Kosten der Einfachheit sehr flexibel.

```
<HTML>
<head><title>The Bookmark Database</title></head>
<BODY>
<H1 align=center>The Bookmark Database</H1>

<!-- DB-Verbindung aufbauen -->
<! msql connect >
<HR NOSHADE><P>

<!-- Datenbank auswaehlen -->
<! msql database bookmarks>

<!-- Datenbankanfrage stellen -->
<! msql query “select url,name from bookmarks
   where url<>’’ order by name” q1>

<!-- Resultat formatiert ausgeben -->
<TABLE BORDER=2><TH> URL <TH> Name
<! msql print_rows q1 “<TR><TD>@q1.1</TD>
   <TD><A HREF=@q1.0>@q1.0</TD></TR>”>

<!-- Resourcen freigeben u. Verbindung schließen -->
<! msql free q1>
<! msql close>
</BODY></HTML>
```

Abb. 4: Web-Anbindung der 2. Generation (W3-mSQL)

- ***Ablage der Seiten***: Während die meisten Produkte mit im normalen Dateisystem abgelegten Informationen (abgesehen von den DB-Daten) arbeiten, speichern andere zusätzlich zu den „normalen“ Daten auch die HTML-Seiten in der DB. Während der erste Ansatz den Vorteil der einfachen Wartung mit Hilfe der üblichen Werkzeuge hat, bietet der zweite die Vorteile eines DBS (siehe „Datentypen“ und „Systemarchitekturen“ in Kap. 5), u. a. erweiterte Suchmöglichkeiten und die Verwaltung von Zugriffsrechten.
- ***Transaktionsdauer***: Von den angebotenen Produkten bieten nur wenige die Möglichkeit, mehrere DB-Operationen zu einer Transaktion zusammenzufassen (Multi-Statement Transaction, MST), um so z. B. durch bedingte Anweisungen eine größere Verarbeitungsflexibilität zu erlangen. Ein Großteil der Produkte unterstützt lediglich Single-Statement Transactions (SST).

Produkte

Mittlerweile existiert eine Reihe von Produkten auf Niveau der zweiten Evolutionsstufe. Neben den direkt vom DBS-Hersteller angebotenen Produkten sind noch zahlreiche von Drittfirmen auf dem Markt erhältlich. In unserer (nicht vollständigen) Übersicht in Abb. 5 haben wir uns auf von DBS-Herstellern angebotene DB-Web-Anbindungen beschränkt. Produktlisten, teilweise auch schon mit Zusatzinformationen, finden sich z. B. bei [Let96, Row96]. Unsere Aufstellung orientiert sich an den von uns oben diskutierten Kriterien. Anzumerken ist, daß es bei allen Produkten möglich ist, während der Abarbeitung noch andere externe Programme aufzurufen, um so weitere Funktionalität einzubinden.

Von uns näher betrachtet wurden W3-mSQL von Hughes [Hug96], DB2WWW von IBM [NS96], das Web DataBlade von Illustra/Informix [Ill95], web.sql von Sybase [Syb96], Adabas D WebDB der Software AG [SAG96], der Oracle WebServer [Ora96] sowie von OODBMS-Anbietern die „Object Design Internet Solution Suite“ [Odi96a] und O2Web [O295].

Produkte	**Programmart[1]**	**Integrationstechnik[2]**	**Seitenablage[3]**	**Transaktionsdauer[4]**
W3-mSQL	C	H	F	S
DB2WWW	C	M	F	M/S
Web DataBlade	C	H	D	S
Sybase web.sql	C/S	H	F	?
Adabas D WebDB	C/S	H	D	?
Oracle WebServer	S	P/H	F	?
ODI ISS	C	H/P	F (D)	M/S
O2Web	C	P	D	S

1: C=CGI S=Server 2: H=HTML M=Makro P=Programm
3: F=File D=Datenbank 4: S=SST M=MST

Abb. 5: Produktübersicht

4.3 Die dritte Generation

Beruhen die beiden ersten Generationen von DB-Web-Anbindungen noch auf dem CGI, so beginnt mit der dritten Generation die Ära von Java. Bedingt durch die Einschränkung der Transaktionsdauer und den Verzicht auf Funktionalität in den beiden ersten Evolutionsstufen wurde nach einer adäquaten Lösung gesucht, um sich von diesen Behinderungen zu lösen. Fündig wurde man 1995 in der von SUN Microsystems entwickelten (Netz-) Programmiersprache Java. Sie bietet die Möglichkeit, in HTML-Seiten eingebundene Java-Programme auf dem WWW-Server bereitzustellen, diese zusammen mit der entsprechenden Seite beim Einladen durch den WWW-Browser in diesen zu übertragen und dort auszuführen (Abb. 3, (1) und (2)). Realisiert man nun DB-Schnittstellen in Java, so stehen alle Möglichkeiten von Client/Server-Systemen zur Verfügung. Dazu öffnet die Java-Anwendung nach dem Programmstart eine Verbindung zum DB-Server, mit dem es über die gesamte Verarbeitungsdauer hinweg in Kontakt steht (Abb. 3, ①a und ②a).

Der Einsatz von Java zur Realisierung einer Web-Datenbankschnittstelle hat gegenüber den auf dem CGI beruhenden Lösungen eine Reihe von Vorteilen:

- ***Transaktionsdauer***: Durch die Möglichkeit, DB-Verbindungen über längere Zeit zu etablieren, können nun auch Anwendungen mit größerer Verarbeitungsdauer realisiert werden.
- ***Komplexe Programme***: Der Browser kann im Gegensatz zu den CGI-Lösungen, wo er nur zur Dateneingabe diente, nun als Ausgabeinstrument einer komplexen Anwendung genutzt werden. Anfrageergebnisse können mit Hilfe der grafischen Möglichkeiten von Java ohne das Anfallen hoher Bildübertragungskosten visualisiert, Bedingungen schon direkt bei der Eingabe und nicht erst bei der Anfrageausführung überprüft und komplexe Berechnungen, wie aus „normalen“ Anwendungen gewöhnt, ausgeführt werden. Entwickelte Programme sind plattformunabhängig.
- ***Direkte Verbindung***: Da die im WWW-Browser laufende Anwendung direkt mit dem DB-Server kommuniziert, können die üblichen Mechanismen (2PC) zur Absicherung einer vom Anwender kontrollierbaren Ablaufsteuerung eingesetzt werden; die diesbezügliche Unsicherheit wie beim CGI entfällt. Daher werden Java-Lösungen auch für „Internet-Banking“ eingesetzt.
- ***Aufhebung des Flaschenhalsproblems***: Alle DB-Aufrufe gehen ohne Umweg direkt an den DB-Server, die Zwischenstufen WWW-Server und CGI-Programm entfallen. Hierdurch wird zum einen der WWW-Server als potentieller Flaschenhals entlastet, zum anderen können die Ausführungszeiten drastisch reduziert werden.

Neben diesen Vorteilen bringt die Nutzung von Java auch mehrere bei den CGI-Lösungen nicht vorhandene Nachteile mit sich:

- ***Längere Ladezeiten***: Ein wesentlicher Nachteil dieser Generation ist die zu Beginn jeder Verarbeitung erforderliche Übertragung der Anwendung einschließlich der DB-Schnittstelle in Form von Java-Klassen vom WWW-Server zum Browser. Für Intranet-Anwendungen und entsprechende Übertragungsbandbreiten fällt dies nicht ins Gewicht, aber bei Nutzung des Internets ergeben sich hier, besonders bei komplexen Applikationen, erhebliche Ladezeiten.
- ***Verlust von HTML***: Da alle Daten mit Hilfe einer Java-Anwendung verarbeitet werden, steht HTML nur noch zur Einbettung der Applikation, nicht aber zur Formulargestaltung und Ergebnisdarstellung zur Verfügung. Formulare, Ergebnistabellen und Grafiken müssen nun in Java programmiert werden, was einen deutlich höheren Aufwand und eine Abkehr von standardisierten Schnittstellen (HTML-Formulare) bedeutet.
- ***Sicherheitsproblematik***: Das auf der Client-Seite im Browser auszuführende Programm wird von einem WWW-Server heruntergeladen. Arbeitet man nicht in einem abgeschirmten Intranet, besteht bei der Übertragung ohne Verschlüsselung die Gefahr der Manipulation, möglicherweise sind auch nicht alle Informationsanbieter vertrauenswürdig.
- ***Einschränkung von Verbindungsmöglichkeiten***: Je nach Systemumgebung und Einstellungen können im Browser ausgeführte Java-Anwendungen Netzwerkverbindungen nur zum Rechner, von dem sie geladen wurden, etablieren. Dies bedeutet, daß WWW-Server und DB-Server auf einem Rechner bereitgestellt werden müssen. Hierdurch sind Kapazitätsengpässe bei der Rechenleistung möglich.
  Um die Einschränkung zu umgehen und zudem den DB-Server vom Internet abzuschirmen, kann auf dem Rechner mit dem WWW-Server ein Kommunikations-Server (Broker) eingerichtet werden, der die Aufrufe des Clients an den zuständigen DB-Server weiterleitet (Abb. 3, ①-④).

Techniken und Produkte

Während bei den Produkten der zweiten Generation in der Regel mit formulierten SQL/OQL-Anfragen gearbeitet wird, bietet sich durch die direkte DB-Verbindung ein breites Spektrum von potentiellen Lösungen:

- ***Java Database Connectivity*** (JDBC, [HC96]): JDBC ist an das weit verbreitete ODBC (Open Database Connectivity) angelehnt und bietet genauso wie das Vorbild die Möglichkeit zur Ausführung von SQL-Befehlen. Neben der Bereitstellung von reinen Java-Treibern besteht eine Alternative in der Nutzung eines JDBC-Aufsatzes auf ODBC (JDBC-ODBC-Bridge), um so bestehende Infrastruktur ausnutzen zu können. SUN bietet in einem Gemeinschaftsprojekt mit INTERSOLV verschiedene JDBC-Software an [Int96], zusätzlich gibt es noch Produkte von WebLogic [Web96] und anderen Anbietern.
- ***Aufrufschnittstelle*** (Call Level Interface, CLI): Ein anderer Ansatz besteht darin, neben C, C++ oder Smalltalk auch für Java eine DBS-spezifische Funktionsschnittstelle bereitzustellen. Durch das starke Wachstum von Java-Anwendungen beeinflußt, wächst die Zahl der auf dem Markt verfügbaren Java-DB-Schnittstellen. Verfügbar sind u. a. MsqlJava [Col96] für mSQL, OCI/Java [Vin96] für Oracle, Confiserie [Poe96] für POET und ObjectStore PSE für Java [Odi96b], eine Java-Bibliothek für persistente Objekte ähnlich der ObjectStore-Technik für C++ und Smalltalk. Desweiteren planen bereits u. a. Informix/Illustra mit dem „Universal Server“ [Inf96] sowie Sybase [Syb95] die Unterstützung von Java. Das in Abb. 6 auszugsweise dargestellte Java-Programm erfüllt dieselbe Aufgabe wie die bisher gezeigten Beispiele. Allerdings müßte nun noch, wie bei den Nachteilen der Java-Lösungen schon angedeutet wurde, die gesamte Ergebnisausgabe programmiert werden.

- ***CORBA***: Das von der OMG spezifizierte CORBA soll die Ortstransparenz von Objekten und die Ausführung von Methoden auf ihnen unterstützen, um verteilte Systeme zu ermöglichen. Java bietet dagegen als eine ideale Ergänzung Plattformunabhängigkeit von erstellten Programmen. Da in CORBA bereits Schnittstellen zu verschiedenen DBS bestehen, kann nun durch die Realisierung eines Java-CORBA-Clients ein DB-Zugriff vom WWW-Browser aus geschaffen werden. SUN unterstützt dies durch die Aufnahme einer entsprechenden Spezifikation in Java 1.1 sowie eigene Produkte. Zudem existieren bereits die Produkte „VisiBroker for Java“ von Visigenic (ehemals Postmodern Computing) [Vis96] und OrbixWeb von Iona [Ion96]. In Abb. 3 ist der Kommunikationsserver dann ein ORB (Object Request Broker), der zwischen der DB-Anwendung im WWW-Browser und einem oder mehreren DB-Servern vermittelt.

```
public class bookmarks extends java.applet.Applet {
 public static void main(String[] args) {
  bookmarks d = new bookmarks();}

 public bookmarks() {
  Msql msql;
  try {
   msql = new Msql();

   // Verbindung aufbauen und DB auwaehlen
   msql.Connect(“agdvs2.informatik.uni-kl.de”, false);
   msql.SelectDB(“bookmarks”);
   // Anfrage stellen
   MsqlResult result =msql.Query(“select name,url
    from bookmarks where url<>'' order by name”);
   String row[];
   while((row = result.FetchRow()) != null) {
    for(int i=0; i<row.length; i++)
   // Ergebnis ausgeben
   }
   msql.Close();
  }
// ...
```

Abb. 6: Web-Anbindung der 3. Generation (MsqlJava)

4.4 Kommende Entwicklungen

Im folgenden wollen wir die von uns erwarteten Entwicklungen vorstellen. Meist sind sie Weiterentwicklungen der in den vorangegangenen Abschnitten beschriebenen Techniken, wobei vor allem die Erweiterung der Funktionalität sowie die Beseitigung der angesprochenen Nachteile im Vordergrund stehen.

ActiveX

Das von Microsoft als Gegenpol zu CORBA geschaffene Gespann aus COM (Component Object Model) bzw. DCOM (Distributed COM) und ActiveX wird durch die Macht des Marktes bei Web-basierten Datenbankschnittstellen an Bedeutung gewinnen. Zudem gibt es bereits Überlegungen, CORBA und DCOM zu integrieren oder zumindest interoperabel zu machen. Mit Hilfe von OLE- (Object Linking and Embedding) bzw. „ActiveX-Controls“ ist es möglich, Applikationen mit entsprechender Unterstützung von (D)COM als Objekte in andere einzubetten und „fernzusteuern“, ohne daß beim Entwurf eines Programmes Schnittstelleninformationen über das andere vorhanden sein müssen (Binärkompatibilität). Neben Textverarbeitungs- und Tabellenkalkulationsobjekten können natürlich nun auch Datenbank-„Frontends“ in ActiveX-fähige WWW-Browser eingebunden und so eine Datenbankanbindung geschaffen werden. Hierzu steht auch das „ActiveX Object Interface“ (auch „OLE DB“) zur Verfügung, das auf der Basis von COM für eine DB-Anbindung sorgt [Rau96].

Java

Zwar bieten die Java-Anbindungen an DBS deutliche Vorteile bezüglich der zur Verfügung stehenden Anwendungsmöglichkeiten, allerdings bedürfen die Ladezeiten, die Sicherheitsproblematik und die damit zusammenhängende Einschränkung der Verbindungsmöglichkeiten noch einer Nachbesserung. Eine Lösung scheint das im Rahmen von Java 1.1 spezifizierte „Java

Electronic Commerce Framework“ (JECF, [Jav96a]) zu bringen, beide werden zum Jahresanfang 1997 erwartet.

In der neuen Java-Version werden die Sicherheitsmaßnahmen erheblich erweitert und u. a. Kryptographie, digitale Signaturen, Prüfsummen und signierte Applets eingeführt (Java Security API). Das JECF baut auf diesen Verfahren auf und bietet noch zusätzliche Möglichkeiten zum Zugriffsschutz, zur Identifikation und Authentisierung. Eine wesentliche Neuerung ist aber die Einführung von „Service-Kassetten“ (service cassettes). Diese bestehen aus einem oder mehreren in Java geschriebenen Software-Modulen, die auf Basis der in Java 1.1 eingeführten Java-Archive komprimiert und signiert in einer Datei zusammengefaßt werden.

Eine Kassette kann, falls sie nicht auf der Client-Seite vorhanden ist, vom Browser beim WWW-Server wie ein normales Java-Programm angefordert werden (Abb. 3, (1) und (2)). Sie wird dann im Gegensatz zur bisherigen Technik nicht in den Adreßraum des Browsers geladen, sondern zuvor dauerhaft im lokalen Dateisystem abgelegt (persistenter Modul-Cache). Nach einer Sicherheitsprüfung wird sie dann in den Browser geladen und kann nun ausgeführt werden. Bei erneuter Nutzung einer Kassette können so die langen Ladezeiten vermieden werden. Stellt man in einer solchen Kassette DB-Funktionalität zur Verfügung, so kann nun wie bisher eine Verbindung zum DB-Server etabliert werden und die eigentliche Verarbeitung beginnen (Abb. 3, ①-④ bzw. ①a und ②a). Da die Herkunft der Programme über Authentisierung abgesichert ist, können gewisse Sicherheitsrestriktionen aufgehoben werden. Daher sind nun auch Verbindungen zu anderen Rechnern als dem WWW-Server möglich, der Kommunikations- bzw. DB-Server und der WWW-Server können auf mehrere Rechner verteilt werden. Das Problem der eingeschränkten Verbindungsmöglichkeiten und der damit verbundenen potentiellen Leistungsengpässe wird somit auch gelöst.

Erweiterungen von HTML/HTTP

Im Bereich der CGI-basierten Ansätze sind Erweiterungen vom Übertragungsprotokoll HTTP, der Dokumentensprache HTML sowie der Server-Funktionalität denkbar. HTTP wurde bereits um längere Verbindungen ergänzt, um so mehrere Dokumentkomponenten auf einmal übertragen zu können. Hierdurch wird der zeitaufwendige Verbindungsaufbau, der bisher für jede einzuladende Komponente nötig war, vermieden. Erweitert man nun die Verbindungsdauer auf den Austausch von mehreren Dokumenten, kann man zusammen mit Verbesserungen der WWW-Server-Funktionalität (z. B. Zustandsverwaltung) längere Transaktionen realisieren.

Ein Teil der benötigten Funktionalität findet sich bereits in Meta-HTML [Fox96], einer Programmiersprache auf Basis von HTML. Ähnlich wie bei sog. „Server-Includes“ muß ein um Meta-HTML erweitertes Hypertext-Dokument vor der Übertragung zum Browser auf der Server-Seite ausgewertet werden. In der umfangreichen Funktionalität enthalten sind auch Befehle zur Zustandsverwaltung sowie der DB-Anbindung.

5. Künftige Anforderungen

Im vorangegangenen Kapitel haben wir die unterschiedlichen Evolutionsstufen von DB-Anbindungen an das WWW vorgestellt und einen Ausblick auf erwartete Entwicklungen geworfen. Innerhalb weniger Monate ist eine Reihe leistungsfähiger Lösungen entstanden, der DB-Zugriff vom WWW aus ist möglich. Allerdings wird das WWW mit seinen neuen Datentypen, den erweiterten Anwendungsmöglichkeiten und den gestiegenen Benutzerzahlen nur unzureichend durch DBS unterstützt. Neue Anforderungen sind entstanden oder zeichnen sich ab. Im folgenden wollen wir kurz die unserer Meinung nach wichtigsten Forschungsschwerpunkte hinsichtlich der Synthese von WWW und DBS skizzieren.

Neue Datentypen

Für die Verwaltung der zur Verfügung zu stellenden Informationen bietet sich bei größeren Datenmengen der Einsatz von DBS an. Allerdings unterstützen „traditionelle“ DBS nicht die Verwaltung von HTML-Seiten und ihrer Komponenten, so daß neue Datentypen für die multimedialen Daten (Text, Bilder, Filme, Musik und auch Programme) benötigt werden. Zwar existieren schon wenige Lösungen, die das einfache Speichern ermöglichen, weitergehende Funktionalität ist in der Regel aber nicht vorhanden. Für die Zukunft sind folgende Anforderungen zu erfüllen:

- ***Dateisystem***: Um zum einen die einfache Migration von auf dem Dateisystem basierenden, bestehenden WWW-Servern hin zu DB-gestützten Lösungen realisieren zu können und zum anderen einfache Verweise innerhalb der DB zu ermöglichen, bietet sich die Bereitstellung einer Verzeichnisstruktur durch das DBS an. Zudem lassen sich Verzeichnisangaben innerhalb einer Adresse durch den Anwender einfacher merken als die meist kryptischen Suchanfragen. Adabas D WebDB [SAG96] bietet bereits eine Verzeichnisstruktur zur Verwaltung der HTML-Daten an.
- ***Intelligente (Volltext-)Suche***: Zu den Vorteilen eines DBS gegenüber einem einfachen Dateisystem gehören auch die (erweiterten) Suchmöglichkeiten, wie z. B. die Volltextsuche. Neben den Fähigkeiten von reinen Textretrieval-Systemen wird noch weitere Funktionalität benötigt, um die HTML-Anweisungen sinnvoll interpretieren und ausnutzen zu können. Möchte man alle gespeicherten Dokumente zum Thema HTML als Ergebnisliste haben, so dürfen nicht alle Dokumente angezeigt werden (alle HTML-Texte beginnen mit der Anweisung <HTML>...). Suche nach Autoren, Schlüsselwörtern, Erstellungsdatum oder nach bestimmten Inhaltsstrukturen (enthält drei Querverweise, Listen etc.) erscheinen angebracht.
- ***Versionierung***: Die Möglichkeit zur (automatischen) Erhaltung und zum Lesen von älteren Dokumentenversionen bringt viele Vorteile mit sich. Zum einen läßt sich verlustfrei editieren, zum anderen können so z. B. ohne großen Aufwand bei ständig aktualisierten Texten auch ältere Versionen (bspw. die Titelseite eines bestimmten Datums bei einer Zeitung) dem Anwender zur Verfügung gestellt werden. Zudem läßt sich mit Hilfe der Versionierung ein großer Datenbestand für den Anwender konsistent halten: Es wird so lange die alte Dokumentversion geliefert, bis alle Seiten und damit auch die Querverweise aktualisiert wurden, dann erst wird auf die neueste Version umgeschaltet. Das Web DataBlade von Illustra [Ill95] bietet Versionierung als „Time Travel“-Option an.
- ***Referentielle Integrität***: Die Wartung der internen Querverweise der Hypertext-Dokumente ist eine sinnvolle Aufgabe, die jedoch wegen des Aufbaus von HTML nur durch eine Analyse der HTML-Seiten möglich ist.
- ***Einfaches Editieren***: Das Editieren der durch das DBS verwalteten Dokumente muß weiterhin mit den normalen Werkzeugen ohne größeren Aufwand möglich sein.

Systeme

Werden WWW-basierte Anwendungen realisiert, die Synchronisation erfordern, entstehen für die DBS eine Reihe neuer Anforderungen:

- ***Lange Transaktionen***: Im Gegensatz zu „normalen“, lokalen Applikationen entstehen durch die Internet-Nutzung teilweise erhebliche Übertragungszeiten zwischen DB-Client und -Server. Kommt dazu noch Benutzeraktion („human in the loop“), so werden aus einfachen Transaktionen Lange Transaktionen mit den entsprechenden Auswirkungen, Leistungsprobleme sind die Folge.

- ***Neue Anwendungen und Synchronisation***: Werden im Rahmen einer Transaktion nicht nur vorgegebene Anwendungsschritte vollzogen, sondern die Möglichkeit zur (mengenorientierten) Anfrageausführung und „Daten-Browsing“ gegeben, hat dies Auswirkungen auf die Leistungsfähigkeit des DBS. Durch das „Browsing“ bzw. die Anfragen können große Teile des Datenbestands für anderen Transaktionen (zum Schreiben) gesperrt werden. Wie bei den Langen Transaktionen entsteht ein geringer Systemdurchsatz durch Deadlocks oder Wartezeiten.
- ***Authentisierung***: Anders als bei lokalen DB-Anwendungen, die nur einem begrenzten Personenkreis offenstehen und bei denen Authentisierung bzw. Autorisierung meist über das Benutzerkennzeichen abgewickelt werden, lassen sich WWW-basierte Applikationen von potentiell jedem Internet-Teilnehmer nutzen. Die vom WWW bekannte Autorisierung über die Eingabe eines Namens und eines Passworts ist für kritische Anwendungen (z. B. Bankgeschäfte über das Internet) nicht ausreichend. Weitergehende Maßnahmen wie z. B. Transaktionsnummern (TANs) sowie Digitale Signaturen werden benötigt. Für das DBS entstehen neue Verwaltungsaufgaben.

Systemarchitekturen

Ein Großteil der CGI-Lösungen der 1. und 2. Generation hat erhebliche Geschwindigkeitseinbußen durch die mehrstufige Aufrufhierarchie. Der WWW-Server wird zum Flaschenhals, hohe Transaktionsraten sind nicht möglich. Die Erweiterung des WWW-Servers mit Hilfe der Server-API beschleunigt nur die Verarbeitung, der WWW-Server bleibt weiterhin ein Nadelöhr. Hauptproblem ist der zentralistische Ansatz, d. h., alle Zugriffe werden über einen WWW- und einen DB-Server abgewickelt. Abhilfe schafft hier nur die Verteilung der Last auf mehrere Server. Während sich statische Informationen relativ einfach auf mehrere WWW-Server verteilen lassen, bereitet eine eventuelle Aufteilung von häufig aktualisierten und in DBS gehaltenen Daten mehr Probleme. Die beiden folgenden (nicht disjunkten) Szenarien zeigen die Hauptprobleme:

- **Szenario 1**: Die DB-Daten werden nicht nur von WWW-Anwendungen, sondern auch von anderen, „traditionellen“ Applikationen (Legacy-Applikationen) genutzt, und das DBS bietet von sich aus keine Möglichkeit zur Verteilung. Möchte man nun die Daten auf mehrere DB-Server aufteilen bzw. entsprechende Replikate anbieten, müssen die (unterschiedlichen) DBS durch eine zusätzliche Schicht (Middleware) zwischen WWW- und DB-Server untereinander synchronisiert und die Kohärenz der Replikate mit dem Original bzw. die des Originaldatenbestands gewährleistet werden (siehe Abb. 7). Lösungen können hierzu u. a. aus dem Bereich von Multi-DBS (MDBS) und Föderierten DBS (FDBS) kommen. Wichtig erscheint vor allem die sinnvolle Integration der Legacy-Systeme.
- **Szenario 2**: Teile eines zentralen Datenbestands müssen weltweit verfügbar sein, andere länderspezifische Informationen nur lokal. Um adäquate Übertragungsbandbreiten für alle Nutzer garantieren zu können, werden daher alle benötigten Daten jeweils lokal, zum Teil als Replikat des globalen Datenbestands, gehalten. Zur Gewährleistung der Datenkohärenz

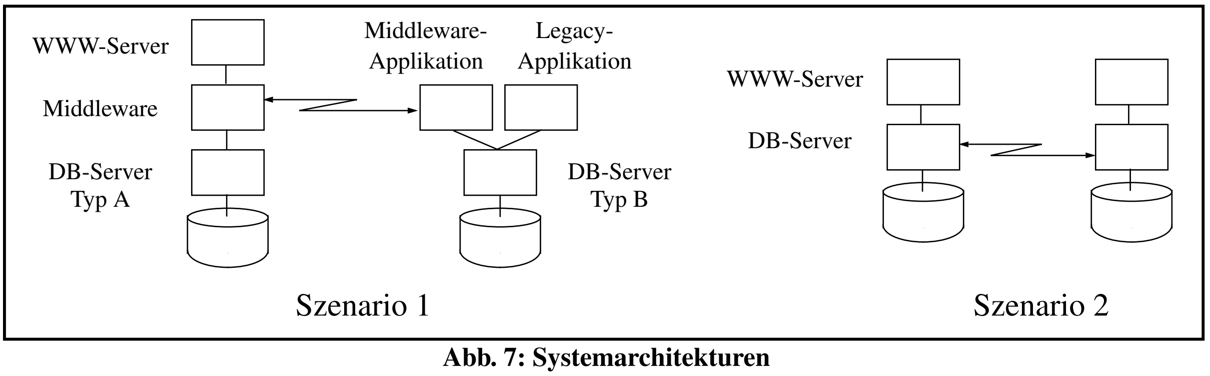

Abb. 7: Systemarchitekturen

wird ein Verteiltes DBS (Distributed DBS, DDBS) benötigt. Anders als bei den meisten der bisherigen Ansätze für DDBS wird jedoch kein lokales Netz, sondern das Internet verwendet. Beim DDBS Mariposa [SAL+96] wird bereits Verteilung auf der Grundlage eines WAN (Wide Area Network) untersucht.

Wichtiger Aspekt bei beiden Szenarien ist die Kooperation von DBS, sei es als Komponenten im Rahmen eines DDBS oder als weitgehend autonome Systeme im Bereich von MDBS und FDBS. Geht man davon aus, daß ein Großteil der „traditionellen“ Daten in (prä-)relationalen Systemen vorliegt, aber für die Verwaltung von HTML-Seiten, wenn überhaupt, überwiegend objektorientierte oder objektrelationale Systeme eingesetzt werden, so ist besonders die Kooperation zwischen den einzelnen „Generationen“ gefordert.

Anwendungen

Neben der Bereitstellung adäquater, leistungsfähiger DBS und entsprechender Schnittstellen zum WWW gilt es nun auch, das WWW als Basis für Anwendungen zu nutzen. Wegen der Plattformunabhängigkeit der Benutzerschnittstelle bieten sich besonders Anwendungen aus einem verteilten, heterogenen Umfeld an. Hierzu gehören neben Enterprise-Information-Systemen (EIS) auch Workflow-Management-Systeme (WFMS), sogar über Unternehmensgrenzen hinweg („Extranet“), und andere Gruppenarbeitswerkzeuge. Steht bei ersteren vor allem die selektive, d. h. benutzerspezifische, Informationsbereitstellung im Vordergrund, ist die Integration von Anwendungslogik und die Kooperation in einem verteilten Umfeld bei WFMS Hauptaufgabe zukünftiger Entwicklungen. Kundenbestellungen und die Abfrage des Bearbeitungsstands per WWW sowie Benachrichtigungen des Kunden per Email können dabei in die Funktionalität der Systeme aufgenommen werden.

Komplexe Systeme lassen sich durch neue Techniken bei der Entwicklung in voneinander getrennte Teilkomponenten aufteilen. Durch die durch ActiveX bzw. (D)COM und Java Beans [Jav96b] gegebenen Möglichkeiten der Software-Entwicklung können nun Teilkomponenten beinahe unabhängig voneinander entwickelt werden. Java Beans definiert ein Komponentenmodell (Component Model), das erweiterte und auch dynamischere Interaktion erlauben soll und COM bzw. DCOM vom Grundprinzip her ähnlich ist. Zu (D)COM und OpenDoc sollen entsprechende Übergänge geschaffen werden, um in Java geschriebene Software-Komponenten auch von anderen Programmen aus nutzen zu können. Es ist nun möglich, eine komplexe Anwendung aus verschiedenen, voneinander unabhängigen Software-Komponenten zusammenzusetzen, man spricht auch von „ComponentWare“. Wird die eigentliche Applikation in einer plattformunabhängigen Sprache entwickelt bzw. in einen plattformunabhängigen Kontext (HTML) eingebettet, so lassen sich auf relativ einfache Art und Weise verteilte Anwendungen in einem heterogenen Umfeld realisieren. Erste Schritte auf dem Weg zur „ComponentWare“ sind das Erkennen der logischen Komponenten der bestehenden sowie neuer Systeme und dann die Realisierung entsprechender Funktionsmodule. Anstelle einer großen monolithischen Anwendung bzw. eines Systems kann nun aus den einzelnen Bausteinen, basierend auf dem Client/Server-Prinzip, ein den individuellen Anforderungen gerecht werdendes Gesamtsystem geschaffen werden.

6. Zusammenfassung und Ausblick

In diesem Beitrag haben wir die unterschiedlichen Techniken der DB-Anbindung an das WWW vorgestellt. Eine optimale Lösung existiert bislang nicht, es gibt einen „Tradeoff“ zwischen der Mächtigkeit der Ansätze und der Einfachheit der Anwendungserstellung. Wünschenswert ist hier eine Lösung mit der Mächtigkeit von Java in Verbindung mit der Einfachheit von HTML. Als Entwicklungstrend ist daher auch die Ergänzung bestehender CGI-basierter Produkte um weitere Funktionalität festzustellen. Es gilt, die Zustandslosigkeit von HTTP zu überwinden, um auch Anwendungen mit längeren Transaktionen zu ermöglichen. In diesem Punkt sind eindeutig die Lösungen auf Basis von Java im Vorteil, da sie eine echte Client/Server-Umgebung realisieren können. Abzuwarten bleibt, wie gut die bisherigen Nachteile von Java durch die Version 1.1 behoben werden. In Zukunft wird auch ActiveX mit (D)COM eine größere Rolle spielen, besonders bei Intranet-Anwendungen.

Während schon mehr oder weniger geeignete Lösungen für die Anbindung bestehender DBS existieren, besteht bei den DBS bezüglich der WWW-Unterstützung noch größerer Forschungsbedarf. Für die Verwaltung von HTML-Dokumenten und ihrer multimedialen Komponenten mit Hilfe eines DBS werden geeignete Datentypen benötigt. Die Koexistenz von neuen WWW-basierten und Legacy-Systemen ist eine Herausforderung, die individueller Lösungen bedarf. Auch die Verwaltung und Anbindung der Daten global operierender und daher weltweit vertretender Unternehmen auf Basis von Intra- und Internet bringt interessante Forschungsaspekte im Bereich der DDBS und MDBS mit sich. Die Plattformunabhängigkeit des WWW mit HTML und Java verdeckt die Heterogenität auf der Benutzerseite und schafft gerade damit ganz neue Möglichkeiten für DB-gestützte Anwendungen.

Danksagungen

Mein Dank gilt Herrn Prof. Dr. Th. Härder für die hilfreichen Hinweise und Anregungen bei der Erstellung dieses Beitrags sowie den anonymen Gutachtern für die konstruktive Kritik zu einer früheren Fassung.

7. Literatur

[Cal96] Cailliau, R.: *A Little History of the World Wide Web*, WWW-Konsortium, http://www.w3.org/pub/WWW/History.html, 1996.

[CGI95] *The CGI Specification*, Version 1.1, University of Illinois, Urbana-Champaign, http://hoohoo.ncsa.uiuc.edu/cgi/interface.html, 1995.

[Col96] Collins, D.: *An MsqlJava Tutorial*, University of Queensland, http://www.minmet.uq.oz.au/msqljava/tutorial.html, 1996.

[FKK96] Freier, A. O., Karlton, P., Kocher, P. C. : *The SSL Protocol, Version 3.0 - Internet Draft*, Netscape Communications, http://home.netscape.com/eng/ssl3/ssl-toc.html, Mai 1996.

[Fox96] Fox, B. J.: *Meta-HTML: A Dynamic Programming Language for WWW Applications*, Universal Access Inc., http://www.metahtml.com/meta-html/manifesto.html, 1996.

[GJS96] Gosling, J., Joy, B., Steele, G.: *The Java Language Specification*, Addison-Wesley, http://java.sun.com:80/doc/language_specification.html, 1996.

[HC96] Hamilton, G., Cattell, R.: *JDBC: A Java SQL API*, Version 1.10, SUN Microsystems Computer Company, ftp://splash.javasoft.com/pub/jdbc-spec-0110.ps, Oktober 1996.

[Hug96] Hughes, D. J.: *W3-mSQL*, Hughes Technologies Pty. Ltd., http://Hughes.com.au/product/w3-msql/manual-2/w3-msql.htm, 1996.

[Ill95] *Illustra Web DataBlade User's Guide*, Release 2.1, Illustra Information Technologies Inc., November 1995.

[Inf96] *Informix and Illustra Merge to Create Universal Server*, White Paper, Informix Software Inc., http://www.informix.com/informix/corpinfo/zines/whitpprs/illustra/ifxillus.htm, Februar 1996.

[Int96] *INTERSOLV and Sun Partnership Bridges the JDBC-ODBC Gap*, INTERSOLV Inc., http://www.intersolv.com/programs/sweet_white.htm, 1996.

[Ion96] *Orbix, distributed object technology - White Paper*, IONA Technologies Ltd., http://www.iona.com/Orbix/Java/index.html, Februar 1996.

[Jav96a] *White Paper: The Java Electronic Commerce Framework (JECF)*, Sun Microsystems Computer Company, http://java.sun.com/products/commerce/doc.white_paper.html, 1996.

[Jav96b] *Java Beans: A Component Architecture for Java*, Sun Microsystems Computer Company, http://splash.javasoft.com/beans/WhitePaper.html, 1996.

[Let96] Letovsky, S. I.: *Web-Database Gateways*, http://gdbdoc.gdb.org/letovsky/genera/dbgw.html, 1996.

[NS96] Nguyen, T., Srinivasan, V.: *Accessing Relational Databases from the World Wide Web*, in: Jagadish, H. V., Mumick, I. S. (Eds.): Proc. of SIGMOD Conf. 1996, Montreal, Canada, S. 529-540, 1996.

[O295] *O2Web - Building a WWW service on top of the O2 database system*, O2 Technology, http://www.o2tech.com/, 1995.

[Odi96a] *Die Internet Solution Suite von Object Design*, Object Design Software GmbH, http://www.odi.com/germany/intersol.htm, 1996.

[Odi96b] *ObjectStore PSE and PSE Pro for Java Release 1.0 Beta 3 Documentation*, Object Design Inc., http://www.odi.com/products/pse/doc/index.html, 1996.

[Ora96] *Oracle WebServer 2.0 - Technical Note*, Oracle Corporation, http://www.oracle.com/products/websystem/webserver/pdf/technote.pdf, März 1996.

[Poe96] *Confiserie: The POET Java Gateway - An Overview*, POET Software, http://developer.poet.de/Confiserie/doc/Overview.html, 1996.

[Rau96] Rauch, S.: *Talk to Any Database the COM Way Using the OLE DB Interface*, in: *Microsoft Systems Journal,* Volume 11, Nr. 7, S. 19-38, Miller Freeman Inc., http://www.microsoft.com/oledb/prodinfo/msjrauch/rauch.htm, Juli 1996.

[Row96] Rowe, J.: *Existing Products for All Platforms*, NASA Langley Research Center, http://cscsun1.larc.nasa.gov/~beowulf/db/all_products.html, 1996.

[RS95] Rescorla, E., Schiffman, A.: *The Secure HyperText Transfer Protocol - Internet Draft*, Enterprise Integration Technologies, http://www.eit.com/projects/s-http/draft-ietf-wts-shttp-00.txt, Juli 1995.

[SAG96] *Entire Broker - Middleware-Technologie für Internet/Intranet-Einsatz*, Software AG, http://www.softwareag.com/sagd/passage/broker.htm, 1996.

[SAL+96] Stonebraker, M., Aoki, P. M., Litwin, W., Pfeffer, A., Sah, A., Sidell, J., Staelin, C., Yu, A.: *Mariposa: A Wide-Area Distributed Database System*, in: VLDB Journal 5(1), S. 48-63, Springer Verlag, http://epoch.CS.Berkeley.EDU:8000/mariposa/papers/s2k-95-63.ps, Januar 1996.

[SET96] Mastercard, Visa: *Secure Electronic Transaction (SET) Specification - Book 3: Formal Protocol Definition, Draft for testing*, http://www.visa.com/cgi-bin/vee/sf/set/setprot.html?2+0, August 1996.

[Syb95] *Sybase "Open for business on the net": Company announces powerful integration of databases with internet web sites*, Sybase Inc., http://www.sybase.com/inc/corpinfo/press/press95/951206a.html, 1995.

[Syb96] „Sybase web.sql Programmer's Guide", Sybase Inc., http://www.sybase.com/products/internet/websql/docs/pguide/pguide1.htm, 1996.

[Vin96] *OCI/Java Gateway - Product Information*, Vincent Engineering, http://www.vincent.se/Products/OCIJavaGateway/OCIJavaGateway.html, 1996.

[Vis96] *VisiBroker for Java*, Visigenic Software Inc., http://www.visigenic.com/prod/vbjpd.html, 1996.

[Web96] *WebLogic jdbcKona products*, WebLogic Inc., http://www.weblogic.com/products/jdbckona_main.html, 1996.