

DFN-Arbeitskreis Informationsdienste

Datenbankanbindung ans WWW mittels JAVA - Ein Erfahrungsbericht

Referent: Michael Hurler
FG Datenverwaltungssysteme, FB Informatik
Professor A. Buchmann
Technische Universität Darmstadt
(<buchmann, hurler>@dvs1.informatik.tu-darmstadt.de)

Praktikum zur DB-Anbindung

- **Ziel:**
Aufzeigen von Vorgehensweisen und Problemen der Anbindung von Datenbanken an das Internet. Hier unter Verwendung von Java.
- **Infrastruktur**
 - Sun Solaris (Sun Sparcstations)
 - Informix Online
 - Java Development Kit
 - Javafähiger Internet Browser (Java v1.1)
 - Internet

Java, das WWW und die Sicherheit

- Java-Programme im WWW (sog. Applets) dürfen nicht auf die Hardware / das Betriebssystem des Rechners zugreifen, auf dem sie ausgeführt werden (Sandkastenprinzip)
- Zugriff von Applets auf das Betriebssystem durch Signieren mit “beglaubigten Unterschriften”
- Signaturen verfolgen Prinzip von Trusted Authorities
- Offiziell sind Signaturen in Deutschland nur sehr schwer zu erhalten und **teuer** (ca. \$1000 p.a.)
- Verwendung von Signaturen auch im Intranet kaum zu vermeiden, da Standard-Browser sich bisher nicht entsprechend konfigurieren lassen

Vorarbeiten I

Java ↔ Informix Online

- Standardinterface: JDBC
- JDBC an Microsofts ODBC angelehnt
- Mehrere Arten von JDBC-Datenbanktreibern, u.a. Native-Java JDBC-Treiber und die JDBC-ODBC-Bridge
- Native-Java JDBC-Treiber bieten Vorteil der Plattformunabhängigkeit und des Sandkastens (keine Signaturen notwendig)
- JDBC-ODBC-Bridge: nutzt breite Basis an ODBC-Treibern, ist aber plattformabhängig und greift auf Betriebssystem zu (Signaturen notwendig und Turn-around-Zeiten bei der Entwicklung hoch)
- Zum Zeitpunkt des Praktikums kein Native-Treiber von Informix oder anderen Herstellern erhältlich

Vorarbeiten II

Klassenbibliothek zur DB-Anbindung

- **Ziel:** vereinfachte und schnelle DB-Anwendungsentwicklung durch Abstraktion von JDBC-Schnittstelle
- Unterstützung durch Objektorientierung Javas
- Komponenten zur Datenanzeige und Datenbearbeitung
- Erweiterbarkeit mit neuen Komponenten durch Botschaftenprotokoll zwischen den Komponenten
- Vorgehen: Definition von dem, WAS eine Komponente tun soll und welche Komponenten “zusammengehören”
- Zukunft: Verbesserungen / Erweiterungen i.B.a. Plattform-unabhängigkeit, Funktionalität und Automatisierung

Probleme

- “Plattformunabhängigkeit”
 - Unterschiede zwischen Java-Portierungen beseitigen (beruhen i.d.R. auf Fehlern in den Portierungen)
 - Größtes Problem: die Bildschirmausgabe (AWT)
 - Selbst bei korrekter Implementierung des AWT komplizierte plattformunabhängige Programmierung
- Performance !!
- Verfügbarkeit von Native-Java JDBC-Treibern
- Browserunterstützung (Netscape, Microsoft, Sun)
- Preiswerte & schnelle Verfügbarkeit von Signaturen in Deutschland

Die Zukunft

- **Java 1.2 & die Java Foundation Classes (JFC / Swing)** als Erweiterung des AWT
 - sehr leistungsfähig
 - weitestgehend plattformunabhängig
- **Deutliche allgemeine Performancesteigerung** nötig
- **Browser** müssen schneller die Verbesserungen im JDK unterstützen (“Java Activator”)
- **Die Technologie** muß endlich erwachsen werden (lt. Sun schon 6 Jahre alt) und die Erwartungen, die sie geweckt hat, erfüllen