

in einer Interface-Spezifikation sich an allen Punkten eines großen Programms, wo dieses Interface benützt wird, auswirken kann (hier nur einige Negative angeführt). Mit Java wurde nur ein kleiner Schritt in Richtung früher Fehlerbeseitigung gegangen.

Component-Konzept: Es ist bekannt, daß Software-Entwicklungskosten je Funktion exponentiell mit der Codegröße steigen: es kostet mehr als doppelt so viel, doppelt so viel Code zu schreiben. Die Java Bean-Technik mag hier eine Verbesserung sein. Nichtsdestoweniger versprechen andere Techniken wie ActiveX, CORBA und OpenDoc ähnliche Verbesserungen; die Java Lösung an sich ist um nichts besser als jene.

Lebenszykluszeit: Applikationen müssen sich synchron mit dem Internet ändern, also ungefähr innerhalb von 18 Monaten. Dies begrenzt die Größe eines neuen Systems, seine Funktionalität oder auch beides. Java hilft wenig im Sinne einer Anpassung an Internet-Zeitabläufe. Kombiniert mit RAD (Rapid Application Development) kann Java allerdings dank seiner Plattform-Neutralität den Aufbau von Systemen innerhalb des Internet-Zyklus ermöglichen - Systemen, die ohne RAD nicht möglich wären.

Komplexität: Heutige Applikationen sind bedeutend komplexer als frühere. Applikationen von morgen werden noch komplexer sein und noch größere Herausforderungen sein. Ganz allgemein steigen die Erwartungen der Öffentlichkeit schneller als die Softwareentwicklung dem nachkommen kann. Java scheint auch kein stärkerer intellektueller Hebel zu sein als Ada, Pascal und so weiter. Noch schlimmer, Java schiebt den intellektuellen Horizont nicht hinaus. So manche andere Sprachen mögen Verbesserungen sein oder auch nicht; sie haben jedoch nicht die gleiche Gelegenheit bekommen, sich zu beweisen, wie Java.

In Java gibt es einiges, dem Beifall zu zollen ist: die Fehlerbehandlung, einfache Vererbung, Interface-Spezifikations-Konstrukte. Außerdem hat Java klugerweise einige wertvolle Eigenschaften von Programmiersprachen vor C/C++ übernommen und als Innovation die Idee des HTML-tagged Applet kultiviert. Fortschritt wurde gemacht (wenn er auch 20 Jahre gebraucht hat). Das aber ist viel zu wenig Fortschritt in einer erwachsen gewordenen Industrie! Java enttäuscht im derzeitigen Entwicklungsstand einfach noch zu sehr.

VBScript

Hans Blocher

VBScript – Visual Basic Scripting Edition – ist eine Untermenge der bekannten Programmiersprache Visual Basic von Microsoft. VBScript wurde als Sprache für WWW-Dokumente entwickelt und sollte leicht, schnell und sicher sein. VBScript erfüllt die beiden ersten Ansprüche, in den Expertenstreit über die Sicherheit von VBScript soll an dieser Stelle aber nicht eingegangen werden.

Zum Leistungsumfang von VBScript gehören Möglichkeiten wie dynamische Generierung von Webseiten oder Überprüfung von Formulareingaben.

Serverseitige Datenverarbeitung

Bei serverseitiger Datenverarbeitung werden von der (Internet-) Station Instruktionen und Daten über das Internet an den Server geschickt. Dort werden diese Informationen verarbeitet und die Ergebnisse wieder an die Station zurückgeschickt, wo sie interpretiert und im Browser angezeigt werden.

Zur Zeit wird im Internet die meiste Datenverarbeitung an den Servern durchgeführt, wo Programme auf Basis von CGI, Java oder Perl ablaufen.

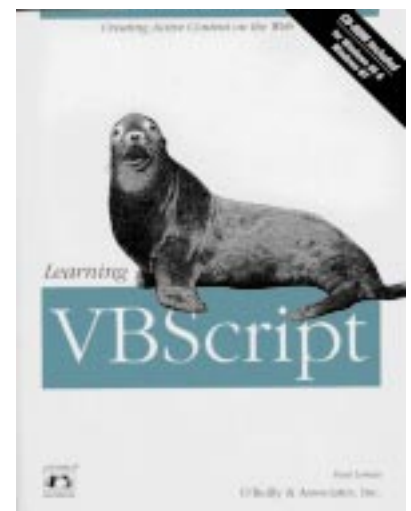
Clientseitige Datenverarbeitung

Bei dieser Art der Datenverarbeitung ist der Browser selbst in der Lage, Instruktionen und Daten zu interpretieren, die sonst an den Server gesandt werden. Die Hauptvorteile sind dabei die Schnelligkeit der Durchführung (es fällt sowohl die langsame Datenübertragung über das Internet weg, als auch die zeitaufwendigen Prozesse, die in den oft überlasteten Servern ablaufen müssen) und die Einsparung an Übertragungsbandbreite.

VBScript ist nun ein Hilfsmittel, mit dem diese Art der Datenverarbeitung unterstützt wird. Es wird als Teil des HTML-Stroms als ASCII-Text zum Browser übertragen, dort mit Hilfe der VBScript Engine kompiliert und im Speicher bereitgehalten. Teile des Programmes können nun auf zwei Arten aufgerufen werden:

- Ereignisse des Browser-Fensters (wie z.B. Anklicken eines Buttons) rufen die dafür definierten Teile auf.
- Bestimmte Programmteile (Funktionen) können von anderen Programmteilen aufgerufen werden.

Dieses Programmbeispiel ist dem englischen Buch "Learning VBScript" von Paul Lomax, Verlag O'Reilly, inkl. CD-Rom mit über 170 Beispielen, entnommen. Dieses Buch ist ein sehr gute Einführung in VBScript, das die wesentlichen Möglichkeiten dieser Script-Sprache bis zu ActiveX-Techniken, dynamische Erzeugung von Webseiten und Überprüfung von Benutzereingaben, abdeckt. Als Zielgruppe



Wahl besteht Hoffnung, daß Java zuletzt ausreifen wird, aber ich kann nicht empfehlen, sich darauf zu verlassen.