[StESHG197] *Ralf Steinmetz, Abdulmotaleb El-Saddik, Markus Schumacher, Christian Herzog, Olav Gröhn Pascal Ilboudo*; **Java in der interaktiven Lehre**; Workshop on Java in Telecommunications, 12.-13. Mai 1997

Workshop on Java in Telecommunication
Deutsche Telekom AG, 12.-13. May 1997
Darmstadt

Java in der interaktiven Lehre

Abdulmotaleb El-Saddik[1], Markus Schumacher[1], Christian Herzog[1],
Olav Gröhn[1], Pascal Ilboudo[1], Ralf Steinmetz[1,2]

[1]Darmstadt University of Technology
Department of Electrical Engineering and Information Technology
Industrial Process and System Communications[1]
Merckstr. 25, D-64283 Darmstadt, Germany
{A.El-Saddik,Markus.Schumacher,Christian.Herzog,Olav.Groehn,
Pascal.Ilboudo,Ralf.Steinmetz}@KOM.th-darmstadt.de

[2]GMD-IPSI Institut für integrierte Publikations- und Integrationssysteme
Dolivo Str.15 64293 Darmstadt
Ralf.Steinmetz@darmstadt.gmd.de

Zusammenfassung

An der Technischen Hochschule Darmstadt wurden im Rahmen eines Projektseminars verschiedene Java-Anwendungen für den Einsatz in der interaktiven Lehre konzipiert und entwickelt. In diesem Beitrag wird die besondere Eignung von Java als objektorientierte Programmiersprache und Entwicklungsumgebung für interaktive Lehranwendungen diskutiert. Anhand eines Beispieles aus einem Hochschulprojekt wird gezeigt, daß eine interaktive Lehre mit Java nicht nur möglich, sondern sinnvoll ist.

1. Diese Arbeit wurde teilweise unterstützt von: Volkswagen-Stiftung, D-30519 Hannover, Germany

1. Einleitung

Unsere Gesellschaft befindet sich im Wandel von einer Industrie- zu einer Informationsgesellschaft, die ein Umdenken in der Informationsdarstellung und -verarbeitung erfordert. Mit konventionellen Ansätzen ist die ständig steigende Informationsmenge nicht zu bewältigen. Der schnelle und effektive Austausch von neuem Wissen und der Zugriff auf bereits vorhandene Daten spielen dabei eine zentrale Rolle.

Die rasante Entwicklung von neuen Technologien und der allgegenwärtige Mangel an Zeit zwingt uns dazu, unsere Wissensaufnahme zu optimieren. Dies gilt nicht nur für die Lehre an Hochschulen, sondern auch für Präsentationen und Lehrgänge der Industrie oder für den Endverbraucher, der eine interaktive Bedienungsanleitung für seinen Videorecorder, sein Telekommunikationsgerät oder seine Waschmaschine benutzt.

Die Informationsaufnahme erfolgt heutzutage überwiegend in schriftlicher oder mündlicher Form. An Universitäten werden manchmal Fachbücher vorgelesen, die der Student zu Hause nachlesen kann, bei Workshops werden Folien aufgelegt, die das Gesagte untermalen sollen, technische Zeichnungen zeigen Anlagen im Stillstand, oder der Käufer eines neuen Fernsehers kämpft sich durch die tabellarische, bis in jedes Detail gehende Bedienungsanleitung.

Ein erster Ansatz der interaktiven Kommunikation ist eine bessere Visualisierung der Informationen. Mit Hilfe von animierten oder bewegten Bildern wird die Informationsaufnahme über das Auge intensiviert. Ein Vortrag mit Demonstrationen und eventuell zusätzlichen Filmen oder Bildern führt dazu, daß der Zuschauende und Zuhörende die Informationen über Augen und Ohren aufnimmt. Außerdem wird durch eine abwechslungsreiche und interessante Informationsdarbietung die Konzentration des Zusehenden/Zuhörers gestärkt.

Um die Aufnahmefähigkeit weiter zu steigern, sollte man die strikte Trennung zwischen Lernendem und Lehrendem lockern, indem man dem Lernenden die Möglichkeit zur Interaktion bietet. Das Wissen soll nicht nur aufgenommen, sondern während des Umgangs und der sofortigen Anwendung verstanden werden. Kleine Kinder lernen instinktiv nach diesem Prinzip, indem sie Dinge anfassen und ausprobieren.

Heute sind Rechner in verschiedenen Ausführungen bereits allgegenwärtig. Universitäten, Firmen und private Anwender verfügen über zahlreiche Rechner, die sowohl intern als auch extern mit anderen Rechnern kommunizieren können. Die Möglichkeiten, die sich durch die Anwendung von Computern in der Lehre ergeben, sind enorm. Moderne Lehr- und Lernprogramme reagieren auf Aktionen des Lernenden, der somit zum Benutzer wird und interaktiv lernt. Für die Entwicklung solcher interaktiver Programme benötigt man adäquate Werkzeuge, die möglichst bereits vorhandene Ressourcen nutzen können. Im folgenden wird auf die Eignung von Java für interaktive Anwendungen in der Lehre eingegangen.

2. Eignung von Java in der Interaktive Lehre

Java ist eine objektorientierte Programmiersprache mit großer Verwandtschaft zu C und C++ [1]. Es handelt sich aber keineswegs nur um eine Sprache für kleine Multime-

dia- oder Internetanwendungen. Java ist eine einfach zu lernenden Programmiersprache, die durchaus für komplexe Anwendungen geeignet ist. Ausgestattet mit zahlreichen Klassenbibliotheken kann man mit Java sowohl auf umfangreiche Datenbanken zugreifen, als auch komplexe, naturwissenschaftliche Probleme simulieren und darstellen. Trotzdem ist Java durch einige Vereinfachungen gegenüber C++ mit relativ wenig Aufwand lernbar. Vor allem für Programmierer, die bereits Erfahrung mit einer objektorientierten Programmiersprache haben, bietet sich Java an. Auch wurde das für den Einsteiger etwas verwirrende und fehlerträchtige Pointer–Konzept weggelassen.

Prinzipiell können Java-Anwendungen als Applet oder als Stand-Alone-Application realisiert werden. Anwendungen im Internet oder Intranet werden meistens als Applet implementiert. Diese sind in der Regel auf WWW-Servern gespeichert und können mit HTML-Seiten [8] verknüpft oder direkt in diese eingebunden werden. Dadurch lassen sich klassische WWW-Dokumente lebendiger und vor allem auch interaktiv gestalten. Außerdem muß der Anwender nicht explizit ein Lernprogramm auf dem Server starten, wenn er beispielsweise beim Lesen eines interaktiven Lehrbuches weiterführende Erläuterungen haben möchte. Allerdings können Java-Applets durch diese Möglichkeiten auch unbemerkt vom Anwender aufgerufen werden. Um zu verhindern, daß diese Applets unkontrollierten Zugriff auf das lokale System oder Netzwerk des Anwenders haben, sind Sicherheitsmechanismen eingebaut und einige Funktionen eingeschränkt worden [9]. Zum Beispiel erlauben Applets keinen Zugriff auf das Dateisystem des Client, um unter anderem zu verhindern, daß Viren eingeschleust werden können. Im Gegensatz dazu lassen sich mit Java-Anwendungen vollwertige Programme entwerfen, die jedoch lokal installiert werden müssen und nicht über einen Browser aufgerufen werden können. Derartige Anwendungen entsprechen herkömmlichen Programmen.

3. Java in der Lehre

Java-Quellcode wird nicht in Maschinencode, sondern in einen Bytecode übersetzt. Dieser Bytecode ist für alle Plattformen gleich, so daß Java-Programme prinzipiell auf jedem Betriebssystem ausführbar sind. Dazu muß das Plattform über einen Java-Interpreter verfügen, der den Bytecode zur Laufzeit interpretiert. Diese Eigenschaft, die Java plattformunabhängig macht, erlaubt es einer breiten Benutzergruppe, Java-Anwendungen in ihrer gewohnten Arbeitsumgebung zu nutzen. Kosten, die bisher durch Portierung von Anwendungen auf die verschiedenen Endsysteme entstanden sind, entfallen. Die Wartungskosten (z.B. Fehlerbehebung) einer Anwendung werden ebenfalls verringert, da Codeänderungen nur einmal anfallen. Lehr- und Informationsanwendungen können so ohne weiteren Portierungsaufwand oder Kenntnis der unterschiedlichsten Plattformen entwickelt werden. Diese Tatsache macht es dem Lehrenden leichter, seine Arbeit einer breiten Öffentlichkeit zur Verfügung zu stellen.

Durch die Netzwerkfähigkeiten von Java können Teile einer Anwendung auf verschiedenen Rechnern abgelegt werden. Zum Beispiel können unterschiedliche Kapitel eines interaktiven Lehrbuches je nach Schwerpunkt der Hochschule auf den Servern der Technischen Universität Darmstadt[1], der Universität Mannheim oder Stuttgart gespei-

1. Die Technische Hochschule Darmstadt wird mit Wirkung vom 1.Okt. 1997 Technische Universität Darmstadt genannt.

chert sein. Die Vorteile dieser verteilten Anwendungen über das Internet oder Intranet sind vielfältig. Obwohl das Knowhow aus verschiedenen Quellen stammt und auch von diesen ständig aktualisiert und verbessert werden kann, ist diese Verteilung für den Anwender vollkommen transparent. Dadurch wird eine vom Ort unabhängige Verwendung von Java-Anwendungen möglich. Der Lernende bestimmt die Bedingungen, unter denen er lernen will. Ein Student kann zum Beispiel eine interaktive Vorlesung zu Hause in Ruhe und beliebig oft nachvollziehen.

Java ist eine neue Programmiersprache. Die Firma SUN, als Entwickler der Programmiersprache, hat bereits bei der Entwicklung von Java einige Multimedia-Konzepte integriert. Es existieren Klassen zum Laden und Abspielen von Audiofiles und zum Anzeigen und Bearbeiten von Bildern in verschiedenen Grafikformaten. Mit dem Abstract Windows Toolkit (AWT) stellt Java einige mehr oder weniger schnell und einfach zu programmierende Möglichkeiten zur Verfügung, um plattformübergreifende, graphische Benutzeroberflächen (GUI) zu entwickeln. Eine einfach und intuitiv zu bedienende graphische Benutzeroberfläche ist eine Grundvoraussetzung für eine interaktive Anwendung in der Lehre. Der Benutzer darf auf keinen Fall gezwungen werden, sich mehr als absolut notwendig mit der Bedienung der Anwendung auseinanderzusetzen.

Die virtuelle Java-Maschine (ein virtueller Rechner, der den Bytecode direkt ausführt) stellt keine großen Anforderungen an die Hardware. Da Java jedoch interpretiert wird, sind derzeit noch nicht die Geschwindigkeiten zu erreichen, wie sie z.B. von C oder C++ Programmen erzielt werden. Wichtig für die interaktive Lehre ist jedoch, daß Java-Programme auf durchschnittlichen Computern akzeptable Reaktionszeiten erreichen. Das ist wichtig, da bisher noch nicht davon ausgegangen werden kann, daß sich auf jedem Schreibtisch ein PC mit Pentium Prozessor oder etwas Vergleichbarem befindet. Diese Frage ist jedoch kein Java-spezifisches Problem. Bei der Wahl einer Zielplattform hatte man in der Softwareentwicklung schon immer die Entscheidung zu treffen, welche minimalen Hardwarevoraussetzungen angestrebt werden. Bei Java-Programmen erübrigt sich die Frage der Zielplattform, die Frage nach der Leistungsfähigkeit bleibt. Die allgemeinen Voraussetzungen, um Java–Anwendungen auszuführen und mit ihnen zu arbeiten, werden derzeit von einem Großteil der vorhandenen Rechner erfüllt. Wenn allerdings umfangreiche mathematische Berechnungen durchzuführen sind, ist mit teils erheblichen Wartezeiten zu rechnen (siehe Abschnitt 6: "Beispiel für die Anwendung in der Lehre").

Zur Zeit spricht noch ein wichtiger Punkt für die Verwendung von Java in der Lehre: Java ist zur Zeit noch kostenlos. Das macht es gerade für Anwendungen sowohl im Hochschul- als auch im privaten Bereich attraktiv. Ob dieser Zustand anhält, ist nicht sicher, der Trend geht aber immer mehr dahin, Java in vorhandene Browser oder Betriebssysteme zu integrieren. Dem Anwender werden Java–Applets somit weiterhin kostenlos zur Verfügung stehen.

4. Anpassung aktiver Elemente an die Gegebenheiten des Lernenden

Wichtig im Zusammenhang mit der Nutzung aktiver Elemente der Java-Applets ist deren Anpassbarkeit an die Gegebenheiten des Lernenden. Man kann mit einem Applet die Komplexität der Lehrinhalte dem Wissen des Anwenders anpassen (z.B. Einführung, Fortgeschrittene, Experte). Darüber hinaus sollte auf die technischen Vorausset-

zungen des Benutzers Rücksicht genommen werden. Im folgenden Beispiel wurde eine Anpassung der Bildschirmgröße angeboten.

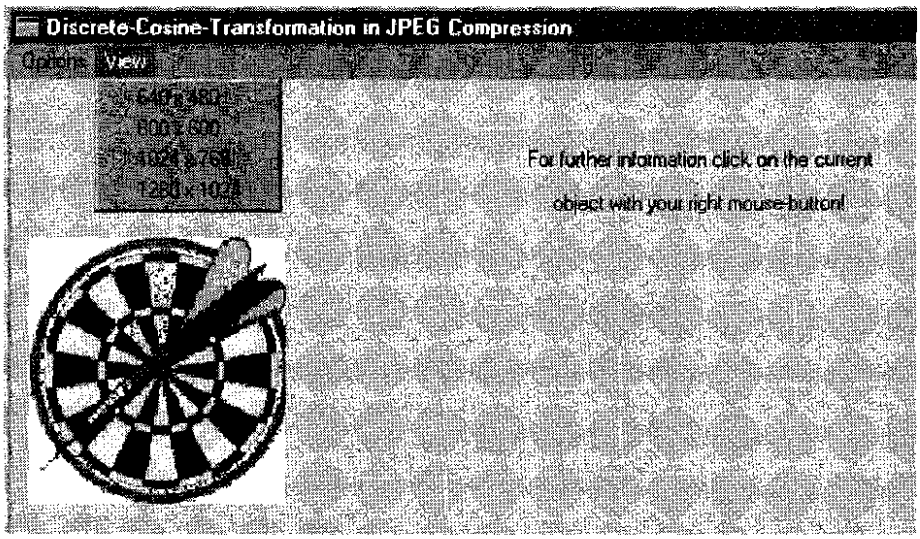

Abbildung 1: Auswahl der Bildschirmgröße

5. Java im Vergleich zu CGI

Für eine dynamische und interaktive Informationsverarbeitung in heterogenen Firmen- oder Universitätsnetzen oder dem Internet gibt es bereits seit einiger Zeit verschiedene Ansätze und Anwendungen [2]. Einer der wichtigsten ist die als Common-Gateway-Interface (CGI) bekannte Lösung [3]. CGI ist als Bestandteil in den Web–Server integriert und kann auf diesem beliebig komplexe Programme aufrufen. Der große Unterschied zu Java besteht darin, daß CGI–Programme auf dem Server ausgeführt werden und auch dort Rechenkapazität beanspruchen, wohingegen Java – Applets vollständig auf den Client geladen werden und dort ausgeführt werden. Dies entlastet nicht nur den Server, sondern auch das dazwischen liegende Netz. CGI–Lösungen erfordern für jeden Benutzereingriff, auf den reagiert werden soll, einen Verbindungsaufbau und -abbau, der jedesmal eine Menge an Overhead beinhaltet. Der Client schickt eine Anforderung an den CGI–Server in Form einer HTTP-Anfrage [10], dieser führt die entsprechenden Programme aus und sendet das Ergebnis (beispielsweise eine dynamisch generierte WWW-Seite) an den Client zurück. Mit der Anzahl der Benutzereingriffe steigt also die Belastung des Servers und die Bearbeitungszeit. Außerdem kann es durch Netzüberlastungen zu langen Antwortzeiten kommen. Bei Java-Anwendungen hingegen wird, wie bereits erwähnt, das vollständige Java-Applet auf den Clientrechner geladen. Dort kann es autark ausgeführt werden und ist nicht von der Netz- oder Serverbelastung abhängig, außer wenn zusätzliche Daten, Java-Klassen oder Remote-Kommunikation z.B. mit einer Datenbank benötigt werden. Der einmaligen längeren Ladezeit eines Java–Applets steht die schnellere Ausführungszeit auf Clientseite gegenüber. Erstaunlicherweise sind auch sehr komplexe Java–Anwendungen mit kleinen (ca. 100 KBytes) Applets möglich.

6. Beispiel für die Anwendung in der Lehre

An der TU Darmstadt wurden im Wintersemester 1996/97 im Rahmen eines Projektseminars am Lehrstuhl von Professor Dr. Ralf Steinmetz drei Anwendungen zur interaktiven Lehre im Bereich Multimedia- und Kommunikationsnetzen entwickelt. Dabei sollte der Verbindungsaufbau und -abbau mit dem im Internet weit verbreiteten Transportprotokoll TCP [11], das Sliding–Window Protokoll [12] zur Flußkontrolle in Datennetzen und die Diskrete Cosinus Transformation (DCT), die unter anderem bei JPEG-Komprimierungsverfahren [13] zum Einsatz kommt, interaktiv und leicht verständlich dargestellt werden. Diese Anwendungen werden in Kürze im WWW auf der Homepage [4] des Fachgebietes verfügbar sein.

Die DCT ist, wie bereits erwähnt, das Kernstück des weitverbreiteten JPEG-Komprimierungsverfahren. Viele Studenten kannten zwar den Begriff JPEG, jedoch nicht die Funktionsweise oder die mathematische Grundlage dieses Verfahrens. Aufgabe war es, eine graphische Computeranwendung zu entwickeln, die flexibel, einfach zu bedienen und leicht portierbar ist und dem Anwender die Möglichkeit zur Interaktion gibt. Diese Aufgabenstellung legte eine Implementierung mit Java nahe, da an der TU Darmstadt unterschiedliche Plattformen existieren und die Programme in der Zukunft für interaktive Lehre im Internet verfügbar gemacht werden sollen. Das Programm sollte dem Benutzer am Beispiel einer Bildkompression mittels DCT erlauben, mit dem Algorithmus zu experimentieren und so die unterschiedlichen Auswirkungen von Parametern kennen zu lernen. Das Projekt wurde schließlich in der Vorlesung "Verteilte Multimedia Systeme - Grundlagen" vorgestellt, um die Akzeptanz bei Studenten zu testen.

In Abbildung 2 ist das Hauptfenster des DCT-Applets zu sehen. Oben links und rechts sind das Originalbild bzw. das komprimierte Bild zu sehen. Darunter ist jeweils ein vom Benutzer frei wählbarer, vergrößerter Bildausschnitt (8x8 Pixel) dargestellt. Durch die Transformation der Bilddaten mit einer diskreten Kosinustransformation entstehen in diesem Fall 64 Frequenzanteile. Die jeweiligen Spektren vor und nach der Komprimierung (eigentlich nur eine Quantisierung) sind der Anschaulichkeit halber als 3D-Balkendiagramm in der Mitte dargestellt.

In der Vorlesung wurden zuvor die theoretischen Überlegungen zur DCT und zu JPEG erläutert und danach das Programm vorgeführt. Die Beschreibung des Programmes und die theoretischen Überlegungen sind aber ebenfalls auf HTML–Seiten dokumentiert, von denen das Programm direkt gestartet werden kann. Durch einen Schieberegler kann der Anwender die Kompressionsrate einstellen und die Auswirkungen auf die Qualität des resultierenden Bildes direkt am Monitor betrachten. Mit der Maus können verschiedene Bildbereiche vergrößert werden, um besondere Bildinhalte zu betrachten (z.B. den Unterschied zwischen weichen und harten Kanten). Einziger Nachteil dieser Java–Anwendung ist die lange Rechenzeit zur Berechnung eines kompletten (120x120 Pixel großen) komprimierten Bildes. Dazu werden ca. 150.000 Iterationen benötigt, die auf einer SUN Ultra Sparc etwa 15s Rechenzeit in Anspruch nehmen. Ein Pentium 133MHz–System benötigt unter Linux für die gleiche Aufgabe etwa 35s Rechenzeit.

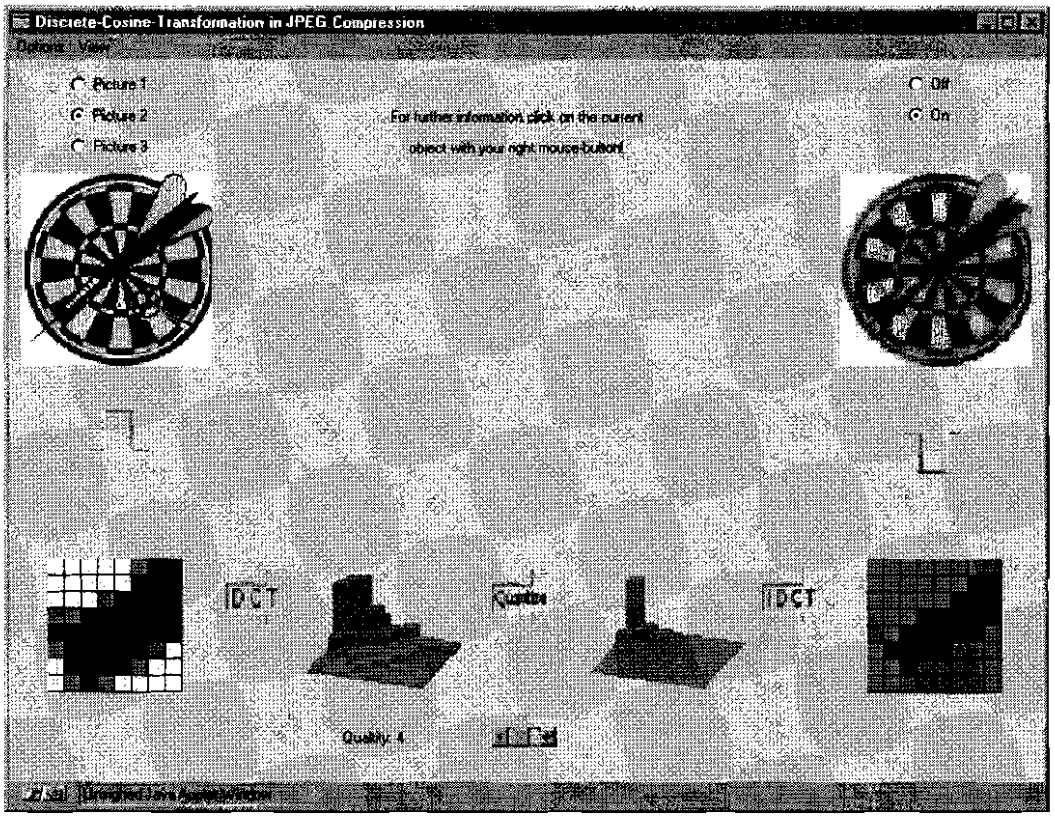

Abbildung 2:Das Hauptfenster des DCT-Applet

Durch diese Interaktivität in der Lehre und die anschaulichere Darstellung läßt sich das Wissen viel besser und interessanter vermitteln. Vielen Studenten ist erst durch dieses Programm klar geworden, warum es bei den bekannten JPEG - Algorithmen zu einer Blockstruktur des komprimierten Bildes bei hohen Kompressionsraten kommt. Die entstandenen Anwendungen haben gezeigt, daß eine interaktive Lehre mit Java nicht nur möglich, sondern auch sinnvoll ist. Die Studenten, die diese Programme testeten, äußerten sich durchweg positiv.

7. Ausblick

Die Projekte, die im Rahmen des Seminars entwickelt wurden, sollen in der Lehre eingesetzt werden. Eine Weiterentwicklung der Programme ist geplant. Weitere Anwendungen sollen in folgenden Semestern implementiert werden.

An Hochschulen wird Java heute als einführende Sprache bereits im Grundstudium gelehrt. In zahlreichen Projekten, Studien- und Diplomarbeiten wird Java als Entwicklungssprache verwendet. Derart ausgebildete Ingenieure und Informatiker tragen ebenfalls zur Verbreitung und Verbesserung von Java bei.

Java ist eine Programmiersprache mit Zukunft. Führende Software-Unternehmen, wie

Microsoft und Netscape, haben Java bereits seit längerem lizensiert und entwickeln eigene Java-Anwendungen. Stardivision [5], ein Hamburger Softwareproduzent, hat bereits jetzt einen Office - Client mit Java entwickelt, der den Desktopversionen lediglich in der Performanz nachsteht. Diese wird sich in Zukunft aber drastisch verbessern durch z.B die Java-Prozessoren, die SUN angekündigt hat. Eindrucksvoll wird hier gezeigt, daß Java dem Stadium einer Experimentalsprache entwachsen ist.

Obwohl Java trotz des geringen Alters bereits sehr robuste Anwendungen ermöglicht, gibt es natürlich einige Probleme und Fehler. Der Großteil der Fehler sollte allerdings mit zukünftigen JDK - Versionen beseitigt sein. Neue, ausgereiftere Entwicklungsumgebungen werden immer mehr Programmierer auf Java umsteigen lassen. Sinnvolle Oberflächen–Design-Tools müssen entwickelt werden, damit man Benutzerschnittstellen nicht mehr mühsam per Hand entwickeln muß.

Da Java interpretiert wird, ist es bisher ca. 30 - 40 mal langsamer als herkömmliche C oder C++ Programme. Eine Geschwindigkeitssteigerung ist also dringend notwendig. Just–In-Time-Compiler (JIT) werden Java schneller machen. Compiler, die es in Maschinencode übersetzen, sind in Entwicklung. Gerade bei rechenintensiven Anwendungen kann so die Leistung der Clients noch intensiver genutzt werden.

8. Literaturverzeichnis

[1] Flanagan, David: "JAVA IN A NUTSHELL", O'Reilly, 1.Auflage, 1996, Deutsche Ausgabe für Java 1.0

[2] CHEOPS, http://stromboli.dia.unisa.it/CHEOPS

[3] Shishir Gundavaran, "CGI Programming on the World Wide Web", O'Reilly & Associates- ISBN: 1-56592-168-2

[4] THD, Industrial Process & System Communications, http://www.th-darmstadt.de/fb/et/ipsk

[5] StarOffice 4.0, http://www.stardivision.de/so40/

[6] Sun Microsystems Inc.: "The Java Tutorial", http://Java.sun.com, 1995

[7] Middendorf, Singer, Strobel, "Java Programmierhandbuch und Reference", dpunkt, 1.Auflage 1996

[8] HTML; Hyper Text Markup Language, http://www.w3c.org/pub/WWW/MarkUp/

[9] Frequently Asked Questions, Applet Security, http://java.sun.com/sfaq/

[10] RFC 2068, HTTP1.1, http://www.w3c.org/pub/WWW/Protocols/

[11] RFC 793, TCP, http://sunsite.auc.dk/RFC/

[12] Andrew Tannenbaum, "Computer Networks", third edition, Printice Hall, 1996 ISBN: 0-13-394248-1

[13] William B. Pennebaker, Joan L. Mitchell, "JPEG Still Image Data Compression Standard", Van Nostrand Reinhold, ISBN: 0-442-01272-1

[14] Ralf Steinmetz, Klara Nahrstedt, "Multimedia: Computing, Communications & Applications", Printice Hall, ISBN: 0-13-324435-0

Proceedings of the
Workshop on Java in Telecommunications
May 12-13, 1997
ITG FA 6.2 System- und Anwendungssoftware
Deutsche Telekom

Anja Harnisch, Joachim Posegga (Eds.)

T · Technologiezentrum