

JAVA - ROBOTER - PIATTFORM

Valerij Volov, E-Mail: v_volov@t-online.de, Internet: www.v-volov.narod.ru

Einleitung

In der technischen Programmierung Literatur ist die anerkannte Äußerung vorhanden, daß Java nicht nur die Programmiersprache oder die Plattform sondern die Philosophie ist [1].

Zweifellos ist Java eine Technologie, die läßt sich oberhalb der üblichen technischen Aufgabe aufzusteigen, bis dem Niveau ihres systematischen Verständnisses, bis dem Niveau des abstrakten Herantretens zu ihrer Lösung. Diese Charakteristik kann man wahrscheinlich in solcher Integrität zu keiner anderen Programmiersprache anwenden.

Aufgrund der allgemeinen und teilweise philosophischen Überlegungen werden in diesem Vortrag die Perspektiven der Java-Technologie in Anwendung zur Robotertechnik betrachtet. Der Vortrag hat die übersichtliche-analytische Eigenart, die sich auf den Materialien der Bücher nach Java-Programmierung und den Materialien einer Reihe der Veröffentlichungen und der Information im Internet nach dieses Thema stützt [3-9].

1. Geschichte und die kurze Charakteristik von Java

Java wurde im 1991 von Mitarbeitern Sun Microsystem vorgehabt, führende von deren Mike Sheridan, James Gosling und Patrick Naughton waren.

Es ist bekannt, daß das ursprüngliche Ziel der Java-Schöpfer die Entwicklung der architekturneutralen Sprache für die Programmierung von Kontroller verschiedener Geräten der Haushaltselektronik war. Deshalb enthält Java in Unterschied von anderen Programmiersprachen die Lösungen, die eine Möglichkeit von Programmen geben, bei verschiedenen elektronischen Umgebungen ausführbar zu werden.

Die Mängel der meisten Sprachen und insbesondere C, C++ bestehen darin, daß die Kompilation nur für eine bestimmte Adresse ausgeführt werden kann. In dieser Beziehung muß man für jeden Kontroller eigenen Compiler herstellen. Das zieht hinter sich die teure Programmentwicklung und keine Mobilität dieser Programme.

Ein Schlüssel, der diese Probleme lösen läßt, ist solche Java-Eigenschaft, daß Java nicht nur eine Programmiersprache sondern auch eine Plattform.

Eine Basis dieser Plattform ist eine virtuelle Maschine, kurz JVM genannt, und APIs (Application Programming Interfaces). JVM führt (interpretiert) den sogenannten Byte Code des Java-Programms. Die

Übersetzung des Quellcodes des Programms in den Byte Code macht die Ausführung des Java-Programms einfacher bei Menge verschiedener Betriebsumgebungen. Die einzelne Bedingung besteht darin, daß JVM für jedes Plattform realisierbar werden muß (siehe Abb.1). Darauf simuliert JVM für Java auf jeder Hardware denselben Rechnerstyp.

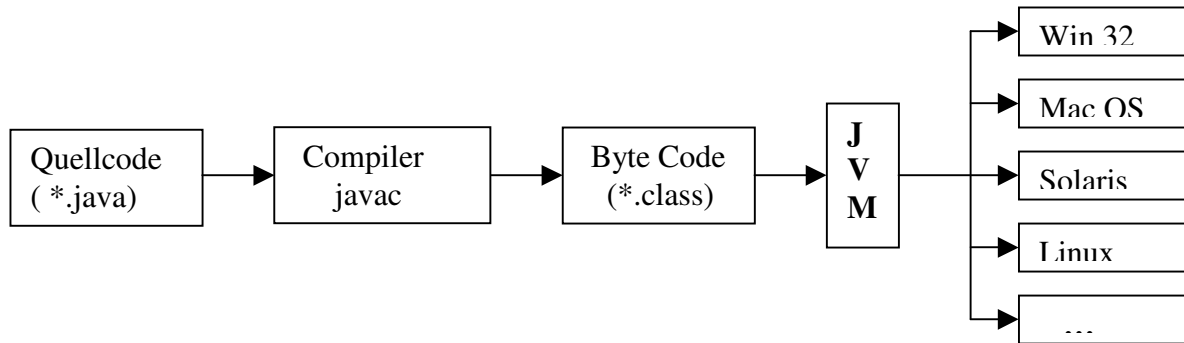


Abb. 1 – Der Vorgang der Vorbereitung und des Startes des Java-Programmes

Wenn man ein Java-Programm zu dem Code irgendeines Kontrollers, als zum Beispiel C++, kompilieren würde, müßen die Versionen dieses Programm für jeden seinen Typ vorhanden sein. Das ist allerdings die schwer zuerledigende Forderung.

Obwohl ein Programm sich langsamer bei der Interpretation im Verhältnis zu der Kompilation in den auszuführenden Code ausführt, ist ein Unterschied zwischen zwei Weisen der Translation nicht so groß. Die Anwendung des Byte-Codes läßt dem ausführenden (run-time) System ein Programm ausführen nicht zu viel langsamer als den ausführende Code. Außerdem gibt es in JVM, als das zusammensetzende Teil, den so genannte JIT-Compiler (Just in Time) für den Byte Code, der den Byte Code in den auszuführende Code während der Ausführung des Programms „Teil für Teil“ nach eine Anfrage kompiliert. Das gibt eine Möglichkeit bei der zweiten Ausführung des Programms sofort zu dem auszuführenden Code überzugehen.

Obwohl die ursprüngliche Aufgabe von Java die Programmierung der Haushaltsgeräten war, ist Java auf eine Vorderstelle der höheren Programmiersprachen wegen der Entwicklung des Internet ausgegangen. Tatsächlich hat sich das dank der Unabhängigkeit dieser Sprache von der Rechnerarchitektur ereignet. Das Internet besteht aus verschiedenen verteilten Bereichen, die mit vielen Rechnertypen, Betriebssystemen ausgefüllt werden. Trotz dieser Unterschied möchten alle Benützer dasselbe Programm ausführen. Eine Verbreitung des Internets in Betracht ziehend schaltete sich ein Fokus von Java an die Internet-Programmierung um. Mit großer Geschwindigkeit in die Welt der Programmiersprachen eingedrungen, sicherte Java die Technologie der Entwicklung der effektiven übertragenden Programme für World Wide Web. Bei diesem realisierte Java erfolgreich folgende Konzepten: Einfach; Objektorientiert; Verteilt und Dynamisch; Interpretiert; Robust; Sicher; Plattformunabhängig; Effizient; Multi-threaded.

2. Java-Technologie in der Robotertechnik

Der Bereich, den die Java-Technologie heute in der Robotertechnik besetzt, kann man viel enger bewerten im Verhältnis zu dem Internet oder sogar den WAP-Telefonen.

Anwendung dieser Technologie in der Robotertechnik bekommt meistens den Impuls der Entwicklung in denjenigen Bereichen, wo der künstliche Intellekt nützliche Markteigenschaften erwirbt.

Grundlegende Roboterfunktionen, die bei diesem die erste Benutzung finden, ist eine Nachahmung einiger menschlicher Funktionen.

Die Java-Technologie schlägt schon eine Reihe API vor, die vielen Bedürfnisse befriedigen können, die bei den Emulation der Funktionen der Rede, des Sehens, der Bewegung usw. entstehen.

Zum Beispiel gibt Java Speech API [3] die Möglichkeit die Befehl-Control-Systeme zu bauen um die menschliche Rede zu erkennen und zu synthetisieren. Und Java Media Framework kann für den Empfang und die Bearbeitung der Bilder angewendet werden.

Ein Vorzug von standarden Java APIs für Robotertechnik vor anderen besteht darin, daß Java-API umfaßt eine Bibliothek an fertigen Softwarekomponenten für beispielweise spezielle Datenstrukturen, grafische Komponenten und Netzwerkprotokolle. Das macht das Verfahren der Programmierung mit Java einfacher. Programmierer kann sich nur an logischen Aspekten einer Aufgabe konzentrieren ohne Besonderheiten und Einzelheiten jedes Systems zu berücksichtigen.

Erfolgreiche Erprobungen einer Integration der Java-Plattform in ein Controlsystem der mobilen Robotern wurden auf den zu programmierenden Controlboxen RCX (Siehe Abb.2), die kleine Rechnern im Gehäuse LEGO [6] vorstellen, durchgeführt.



Abb.2 – Die allgemeine Ansicht des Boxes RCX LEGO

Der Box RCX enthält 8-bit Prozessor von Hitachi (16 MHz), 16 Kb ROM, 32 Kb RAM, 3 Sensorseingänge, 3 Motorausgänge, 5-Symbol LCD-Anzeige und Infrarotschnittstelle.

Drei Ausgänge kann man zu Elektromotoren oder anderen elektromechanischen Mechanismen anschließen. Drei Eingänge kann man mit den verschiedenen Sensoren (Licht, Kontakt, Drehung, Temperatur) anschließen. Das System kann mehr 1000 Operation pro Sekunde ausführen und wird ebenfalls mit dem vielrechnenden Betriebssystem ausgestattet, das gleichzeitig bis 10 Prozessen ausführen kann. Allerdings wird wegen Begrenzen von ROM und RAM die abkürzende Variante der JVM, die hat nur 14 Kb, verwendet. Für die Softwareentwicklung wurde die Plattform Robotics Software Development Kit entwickelt [7].

Noch eine Richtung, die die Entwicklung auf der Basis der Java3D-Technologie bekam, ist die 3D-Modellierung der virtuellen Robotern und die Steuerung ihnen in der virtuellen Umgebung nach dem Internet-Protokoll. In diesem Gebiet kann man gemeinsame Projekte des Institutes von Robotertechnik und des Institutes von Robotern und Mechatronik von DLR bezeichnen. Diese Projekte widmen sich der Entwicklung der virtuellen Modellen des zusteuerenden durch das Internet Roboters ROTEX (Siehe Abb.3) [8].

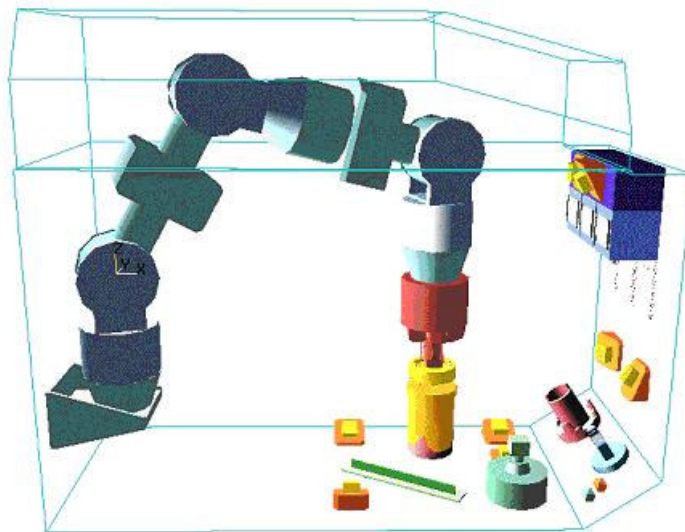


Abb. 3 – Das virtuelle Modell des Roboters ROTEX

Interessante Projekte der Java3D-Modellierung werden im Institut der angewandten Mathematik von Keldisch (Die Abteilung der Mechanik und Steuerungsbewegung) durchgeführt.

Zum Beispiel wurden auf der Basis der Java3D die Methoden der Steuerung in der virtuellen Umgebung mit Verzögerungen entwickelt. Diese Methoden sind effektiv bei der Verwendung des virtuellen Roboters für die Steuerung dem echten Roboter. Diese Systeme wurden für die Roboter PUMA, CRS und „Diligent“ erstellt. (Siehe Abb.4) [9].

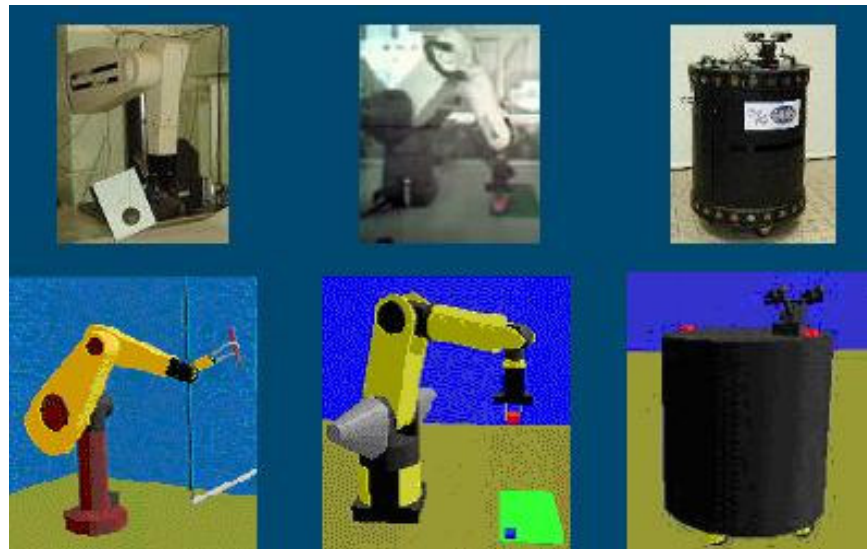


Abb. 4 - Manipulatoren PUMA 560 und CRS A465, der mobile Roboter Nomadic XR4000 "Diligent" (die Reihe von oben, von links nach rechts) und ihre 3D zusteuenden virtuellen Modelle (die Reihe von unten)

Das System für die Steuerung des Roboters (siehe Abb.5) durch das Internet enthält 3 Teile – Roboter, Server und Client. Der Roboter besteht aus dem Manipulator und Controller. Das Server enthält einen Rechner und 2 TV-Kameras IEEE 1394. Für die Steuerung des Roboters „Puma“ benutzt man den Rechner AMD Athlon 1333. Für den Roboter „Diligent“ – Sun Ultra mit Prozessor UltraSPARC. Die Software des Clients begründet sich auf open-sours Java und Java3D, deshalb kann man als Client beliebigen Typ des Rechners oder Betriebssystems benutzen. Der Controller des Roboters verbindet sich mit dem Server durch das Interface RS232. Die Verbindung zwischen dem Client und Server wird über die Standardprotokolle TCP/IP gebaut.

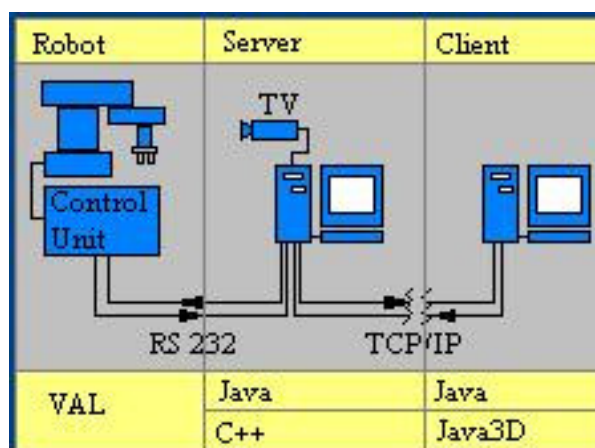


Abb. 5 – Die Architektur des Controlsystemes des Roboters auf der Basis der virtuellen Modelle

Methoden der Steuerung des Roboters bei Verzögerungen, die wegen der technischen Begrenzungen der Geschwindigkeit über die Protokolle TCP/IP im Internet vorhanden sind, können

offensichtlich eine weitere Verwendung bekommen, zum Beispiel für die Steuerung von Raumfahrtobjekte und Planetenfahrzeuge, wo in der Verzögerungen die Begrenzungen dominieren können, die von der Geschwindigkeit des Radiosignals und Weltraumstrecken abhängen. Zum Beispiel kann die Verzögerung für einen Operator bei der Steuerung dem Mondfahrzeug ungefähr 3 s sein, bei der Steuerung dem Marsfahrzeug – ca. 40 min.

3. Perspektiven der Entwicklung der Java-Technologie als einer Plattform in der Robotertechnik

Die letzte Zeit ist der Bereich der Robotertechnik für Entwicklern der Java-Technologie noch mehr attraktiv geworden ist. „Robotertechnik ist der komplizierte Bereich. Die Softwareentwicklern fangen nur an zu verstehen, wie interessante und würdige Aufgaben hier stehen“ - so Dr. Robin Merphi (Perseptual Robotics Laboratory).

Die Grenzen der Java-Plattform in der Robotertechnik erweitern sich. Die Fachleute aus Sun Labs mit den Spezialisten von Nasa und JPL besprechen die Frage der Anwendung der Java-Plattform für neue zukünftige Marsfahrzeuge.

In allerhanden Gebieten der Robotertechnik wird Anwendung der modernen objektorientierten architekturunabhängigen Technologien als aktuell und sehr notwendig. Als ein wesentliche Element der objektorientierten Programmierung ist eine Abstraktion. Das läßt in dem so komplizierten Bereich wie Robotertechnik die Aufgaben der Steuerung der Kompliziertheit durch die Abstraktion lösen. Außerdem lassen solche grundlegende Eigenschaften von Java, als einfach, verteilt und dynamisch, robust, sicher usw. sprechen über die Perspektiven der Entwicklung der effektieven Robotersysteme auf der Basis dieser Technologie.

Heute wird es offensichtlicher, daß die Erforschungen des Weltraumes den Anstrengungen aller Menschheit erfordern.

Die kosmischen Projekte tragen mehr und mehr internationalen Charakter. In diesen Projekten ist normalerweise die Zusammenarbeit verschiedener Teilnehmerländer und die Benutzung verschiedener Produkte zu vermuten. Deshalb kann die Verbindung der Programmeninterfaces als ein Problem wegen der Unterschiede der Architekturen werden und als folglich ist zum Abbruch der Programmenausführung wegen ihrer Zusammenwirkung zu führen.

Zur Erklärung nehmen wir zum Beispiel ein Planetenfahrzeug als ein komplizierte Controlobjekt in der Robotertechnik. Es enthält ein Controlsystem und viele andere, zum Beispiel Telekommunikationsystem, Energieversorgungssystem, Navigationsystem, System technisches Sehens, Sensorensystem und Forschungsgeräte. Alle diese Systeme und Geräte können auf der Basis verschiedener Kontrollere, Prozessors erstellt werden. Sie können verschiedene Betriebssysteme haben. Sogar bei enger Arbeit der Entwicklern dieser Systeme kann die Vereinbarung der elektrischen und

programmen Interfaces zu großen Aufwänden führen. Die Entwicklung und Betreuung des Software nimmt in Anspruch auch zu viel Zeit und Leistung.

Ausgang aus dieser Lage sieht man in diesem Fall durch Anwendung der Java-Technologie, die dieses Problem der Übertragung des Programmencodes auf verschiedenen Plattformen lösen läßt.

Noch eine Richtung der Anwendung der Java-Technologie ist die Modulaufbauten der Roboter am Bord der Weltraumstation. Als wurde oben gesagt, läßt Java ohne überflüssigen Aufwände das Problem der Modulverbindung und Zusammenwirkung verschiedener Roboter lösen. Gleichzeitig realisiert sich das Prinzip des vielrechnenden Systems, denn für ein Programm können die Prozessressource des Modulaufbautes trotz ihrer verschiedenen Architektur überlassen werden.

Noch ein Vorrang der Java-Technologie ist ihre Dauerhaftigkeit. Ihre Vergangenheit formiert ihre Grundlage und bildet für viele Jahre an die Zukunft eine vereinbarende Basis insbesondere in API. Das bedeutet, daß die billigere Modernisierung und die Erweiterung der Möglichkeiten der Roboter-Generation, die Erneuerung des Controlstrukturs durch entweder die Entfernung oder den Ersatz der einigen Teilen des alten Systems (sowohl Hardware als auch Software) möglich ist. Es ist offensichtlich, daß dieses Verfahren die preiswerte Richtung im Verhältnis zu der Variante des Ersatzes des Controlsystems bei der Robotermodernisierung sein könnte.

Zusammenfassung

1. Die grundlegenden Eigenschaften der Java-Technologie begründen sich auf dem objektorientierten Paradigma der modernen Programmiersprachen und der Architekturunabhängigkeit. Deshalb hat Java den Grund als eine perspektive Plattform in der Robotertechnik berücksichtigt zu werden.

2. Der Vorrang, den mit der Java-Technologie in der Robotertechnik angeboten wird, kann man kurz mit solchen Eigenschaften zusammenfassen: architekturunabhängige Unterstützung der vielrechnenden Ressource, Vielstromigkeit der Aufgaben, Mobilität (Übertragung) des Programmes, Möglichkeit der verteilten Ausführung der Prozeduren (Client-Server Technologie), Lösung der komplizierten Aufgaben durch die Methoden der hierarchischen Abstraktion und schließlich die Dauerhaftigkeit der Java-Plattform und zugeprüfte durch die Praktik von Millionen Benutzer ihre Sicherheit.

Literatur und Verweise

1. Helmut Erlenkoetter. Java-Applikationen: Anwendungsprogrammierung mit JFC. Rowohlt Taschenbuch Verlag GmbH, Reinbek bei Hamburg, Dezember 1998.

2. P.Naughton, G.Schild. Java 2: Übersetzung von englisch – SPb.: БХВ-Петербург, 2000.- 1072 s.

3. <http://ru.sun.com/win/java/2001results.html> - SAN, Russland

4. <http://java.sun.com/products/java-media/speech/> - Java Speech API

5. <http://java.sun.com/products/java-media/jmf/2.1.1/> - Java Media Framework API 2.1.1

6. <http://java.sun.com/features/2002/05/robot.html> - Java™ Technology and Lego Mindstorm

Robots

7. <http://developers.sun.com/dev/evangcentral/totallytech/robo-sdk.html> - Java Technology

Robotics Developers Kit

8. <http://www.robotic.dlr.de/VRML/Rotex/#Intro> - Site für die Steuerung dem virtuellen Modell des Weltraumroboters. (Zusammenprojekt des Institutes von Robotertechnik und Institutes von Robotern und Mechatronik von DLR)

9. <http://www.keldysh.ru/pages/i-robotics/rus/rhome.html> - Site von I.Belousov (Institut der angewandten Mathematik von Keldisch (Die Abteilung der Mechanik und Steuerungsbewegung))

Rauenberg, Deutschland, November 2003