

Java - What's All The Excitement About?

Mark Volkmann, Object Computing, Inc. (OCI)

Outline

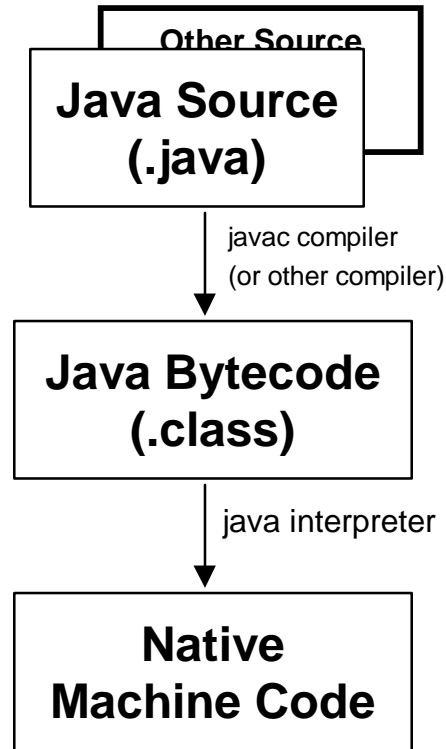
- Java Overview
- Java Security
- Java Performance
- Java Availability
- Future Directions
- Alternatives to Java
- Wrapup
- Appendix A - Lists
- Appendix B - Detailed History

What is Java?

- A new object-oriented language
 - used for internet-based applications AND “normal” applications
- An environment
 - toolset, class libraries, security mechanisms
- Compiled for speed
 - to produce bytecode which is machine language for an imaginary computer called the Java Virtual Machine (JVM)
- Interpreted for portability
 - bytecode is interpreted to native machine code
 - » bytecode is close to the machine code for real machines
 - » interpreting bytecode to machine code is much faster than interpreting from source code
 - **java** interpreter for applications
 - **web browser** or **appletviewer** for applets
- Will be directly supported by most major OSes
 - Apple, Hewlett-Packard, Hitachi, IBM, Microsoft, Novell, SCO, Silicon Graphics, Sun, Tandem

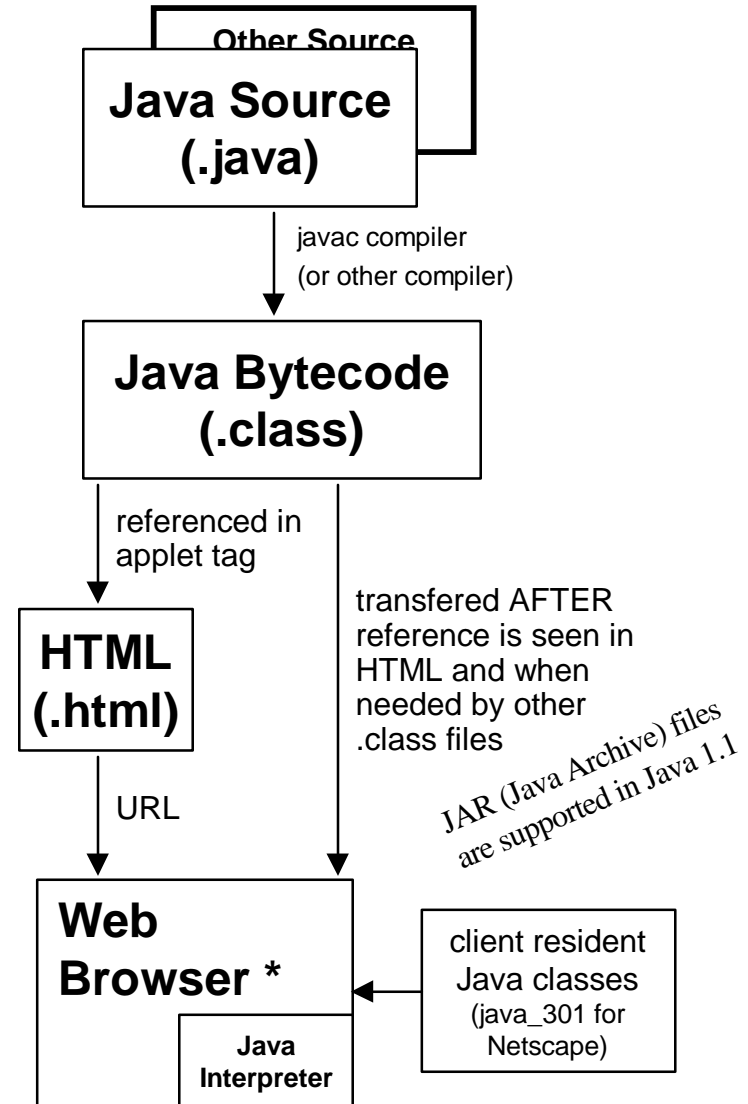
Java Flow

Normal Application



* Java-aware browser or appletviewer

Web-based Application (Applet)

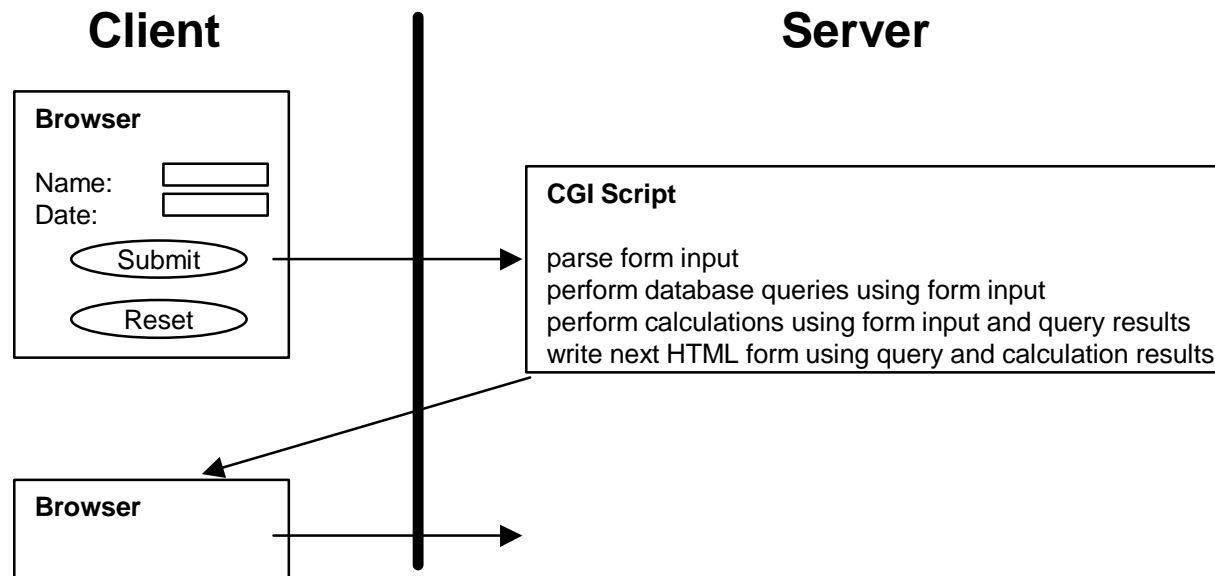


Java Adds Dynamic Content

- Web pages with **static** content don't change
 - ex. pictures of new homes offered by a builder, lists of options, prices
 - created with HyperText Markup Language (HTML)
- Pages with **dynamic** content change
 - ex. colors in pictures are changeable, options can be selected, total price and monthly payments for different interest rates can be calculated
 - dynamic content includes
 - » animation
 - » audio
 - » graphics (not images)
 - » keyboard input
 - » mouse interaction (includes drawing)
 - created with CGI, JavaScript, Java, and plug-ins (ex. Shockwave and VRML)

Common Gateway Interface (CGI)

- One way to achieve dynamic content
- Each CGI script invocation requires a new internet connection to be established
- Work is done on server
- Cannot validate input on client



Web Page Creation

- Past
 - HTML
 - » typical elements
 - header, list, link, image, image map, table, frame
 - » GUI forms
 - text entry(single & multiple line), password entry, checkbox, radio button, file selection, option menu, scrolled list, submit button, reset button
 - CGI
- Present
 - HTML
 - » new authoring tools make this easier
 - CGI
 - » still useful when server-side processing is desired (can also use Java for this)
 - JavaScript
 - » supports simple client-side processing
 - Java
 - » supports complex client-side processing

Origins of Java

- Uses the best features of
 - C++ (syntax)
 - Objective-C (interfaces)
 - Smalltalk & UCSD-Pascal (virtual machine bytecode)
 - others (Eiffel, Lisp, Pascal, Cedar, Self, Beta)
- Quote from Arthur van Hoff
(former Sun employee and one of the main members of the Java team)
 - “We never claimed that Java is new.
It is simply good technology repackaged!”

Sample Java Applets

- Web browser must be Java-aware to run applets
- Nuclear Power Plant simulation at www.ida.liu.se/~her
- Bar and Line Graphs at www.javasoft.com/applets
- ESPN SportsZone at espnet.sportszone.com
- Gamelan at www.gamelan.com
 - library of contributed Java applets
 - source code is included, good way to learn
- Others at www.javasoft.com/applets
 - try [MoleculeViewer/example3.html](#)

Object-Oriented Benefits

- Better software design
 - different way of thinking about software
 - software can more closely models real world problem being addressed
 - domain experts can more easily participate in initial software design
- Easier to maintain
 - problems are easier to isolate
 - changes are more localized
- Easier to extend
 - easy to make extensions to existing functionality
- Less code to write
 - more likely to reuse code from other applications
 - more likely to find commercial code (class libraries) that can be used for portions of an application

Features Important for an Internet Language

- **Secure**
 - discussed later
- **Portable**
 - sizes of primitive data types are the same in all implementations
 - don't have to recompile to run on another platform
- **Fast**
 - compared to scripting languages like Perl, Python, TCL, and JavaScript
 - discussed in more detail later
- **Simple to release**
 - don't have to copy to each client machine
 - latest version downloaded automatically at run-time

Features Important to Developers

- **Object-Oriented** (benefits discussed earlier)
- **Productive**
 - fewer compiles are needed than with C/C++ and makefiles
 - faster development cycle since linking is not required
- **Familiar**
 - syntax resembles C and C++
- **Garbage collection**
 - avoids memory leaks (failing to free memory that will no longer be used)
 - avoids accessing freed memory
 - lack of automated memory management in other languages is a source of memory leaks and bugs
- **Extensible**
 - applications can call functions written in other languages

Aspects of C/C++ That Java Improves Upon

- Time consuming links
- Confusing features
 - multiple inheritance
 - operator overloading
 - templates
- Dangerous features
 - unsafe casts between unrelated types
 - array access w/o bounds checking
 - lack of memory management (garbage collection)
 - » pointer arithmetic
 - » potential for memory leaks
 - » potential for overwriting memory
- Non-OO features

Java Security

- Prevents applets from
 - accessing client files →
 - starting processes on the client
 - using code written in other languages
 - making network connections other than to the applet host
- Security is provided by
 - bytecode verifier
 - class loaders (prevents replacement of JDK classes)
 - SecurityManager class
 - lack of pointer manipulation
- Digitally signed applets
 - support is in JDK 1.1; waiting on support in web browsers
 - users can tell their web browser to trust specified digital signatures
 - trusted applets can have fewer client-side restrictions

cannot read, write, rename, or delete a file
cannot create a directory or list files in one
cannot determine a file's existence, type, timestamp, size

How Fast Is It?

- Currently Java would not typically be used for computationally intensive tasks
- Javasoft is working on
 - more efficient and less disruptive garbage collection
 - continually optimizing interpreter
 - improved thread synchronization manager
- Tests run with JDK 1.0.2 under Solaris
 - **Bubble Sort** (sorting 6,500 integers)
 - » C++: 33.4 seconds
 - » Java: 311.3 seconds (9.3 times C++)
 - » Perl: 1662 seconds (49.8 times C++)
 - **Math Calculations** (1.2 million square root and log calculations)
 - » C++: 9.6 seconds
 - » Java: 27.8 seconds (2.9 times C++)
 - » Perl: 48.8 seconds (5.1 times C++)

Components of Speed

- The speed of a Java **applet** has three components
 - time to download applet to client
 - time to verify bytecode
 - execution speed once it is there
- Java **applications**
 - are not downloaded from the internet
 - aren't verified by default

Ways to Increase Speed Coming Soon

- Download faster
 - by increasing speed of internet connection
- Eliminate need to download
 - by caching recently downloaded applets at client
- Retain internet connection using improved HTTP protocol
- Run faster
 - Just-In-Time (JIT) compilers
 - » instead of processing bytecode instructions one-at-a-time, interprets entire methods to machine code when called and saves
 - » may optimize if called several times
 - » 10x to 25x speed increase (near speed of non-optimized C)
 - native methods
 - native executables
 - » one approach is to translate bytecode to C then compile and link
- Process Java bytecode in hardware
 - Sun is creating processors to do this

Java Development Kit (JDK)

- Free from Sun (www.javasoft.com)
- Includes class libraries
 - applet, awt, beans, io, lang, math, net, rmi, security, sql, util, & more
- Includes tools
 - javac - compiles Java source files
 - java - applications bytecode interpreter (profiling w/ -prof option)
 - appletviewer - applets bytecode interpreter
 - javadoc - generates HTML class documentation
 - jdb - line-oriented debugger
 - jar - creates Java archive files
 - javah - generates header files for using native methods
 - javap - lists fields and methods in a bytecode file
- Current version is 1.1.2

Availability

- Free JDK Ports
 - Sun - Solaris, Windows 95/NT, MacOS
 - Microsoft - Windows 3.1/95/NT
 - IBM - OS/2, AIX, Windows 3.1
 - Digital - Digital UNIX
 - Hewlett Packard (HP) - HP/UX
 - Silicon Graphics (SGI) - IRIX
 - OSF - HP-UX, Digital Alpha, other UNIX OSes
 - Linux and Amiga ports are also available
- Commercial development environments
 - Asymetrix (Supercede), Borland (JBuilder), IBM VisualAge, Metrowerks (CodeWarrior), Microsoft (Visual J++), Natural Intelligence (Roaster), Rogue Wave (JFactory), Silicon Graphics (Cosmo), Sun (Java Workshop), Symantec (Visual Café), Visix (Vibe), and others

How Sun Profits

- Java development environment
 - Java Workshop
 - written in Java for portability
 - has speed problems
- Licensing Java to software vendors
 - necessary to imbed java interpreter into commercial software
- Specialized Java class libraries
 - examples
 - » 3D modeling classes (related to VRML)
 - » image processing classes
- Expanded market
 - increased demand for their internet related hardware (servers and JavaStations) and software through name association with Java

New Features In Java

- **Object Serialization**
 - allows a graph of objects (an object and all objects reachable from it) to be converted to and from a stream of bytes
 - can write the bytes to a file to achieve persistence
- **Java Database Connectivity (JDBC)**
 - java.sql package supplies classes for accessing relational databases in a common way
 - JDBC-ODBC bridge is included
 - vendors that have pledged support include Borland, Gupta, IBM, Informix, Intersolv, Object Design, Oracle, RogueWave, SAS, SCO, Sybase, Symantec, Visigenic, WebLogic, and more
- **Distributed Objects**
 - CORBA access through Java ORBs such as Visigenic's Visibroker for Java, IONA's OrbixWeb, and Sun's JOE
 - Remote Method Invocation (RMI) for Java-only distributed objects

Microsoft's ActiveX

- Currently Java's only serious competition for providing client-side, web-based applications
 - a development kit composed of specifications and tools for generating internet-based applications
 - » uses Visual Basic (VB) Script (compare to JavaScript)
 - » uses OCXes (OLE Controls) (compare to JavaBeans)
 - OLE stands for Object Linking and Embedding
 - used to make custom GUI widgets/forms and specify their functionality
 - created with Visual Basic and Visual C++ (compare to Java)
- **Pros of Active X**
 - mature graphical GUI builders are available now
 - compatible with other Microsoft products through OLE
 - » Sun is working on allowing Java to do this as part of JavaBeans
 - allows creation of internet applications that utilize OCXs
 - large number of existing commercial OCXs
 - Visual Basic developers can utilize current skills rather than learning a new language and development environment

Microsoft's ActiveX (Cont'd)

- **Cons of Active X**
 - lacks many features of Java
 - » security
 - can execute OS commands and use pointers to access memory so can damage client systems
 - only security will be digital signatures
 - allows users to decide amount of risk to take - no sandbox
 - benefit is that when client machine is damaged you'll know by who
 - » portability
 - doesn't have a platform neutral compiled format
 - currently only works under Microsoft OSES
 - doesn't run on a virtual machine
 - requires a different compiled and linked version of each OCX for each OS
 - web servers will have to send OS specific versions of OCX's to web browsers
 - Software AG in Germany is porting Active X support to UNIX OSES
 - has less industry support than Java
 - » more vendors are making their products compatible with Java than ActiveX

Reasons to Select Java Over Other Languages

- Web-based applications
 - common interface to applications/data inside and outside company
- Portability
 - even for non-web-based applications
 - applications will run on all major platforms
- Distributed Applications
 - RMI is easier and less expensive than CORBA but lacks services
- Networking
 - supports TCP and UDP sockets
 - can access remote data using a variety of protocols

Reasons to Select Java Over Other Languages (Cont'd)

- Multithreading
 - can utilize multiple processors
 - » avoids waiting on a single processor
 - more natural style of coding for some applications
 - » especially simulations
- Object-oriented
 - encourages better software design
 - easier to maintain and extend
 - reuse
- Garbage collection
 - software is less prone to errors
- Productivity
 - fewer source files need to be compiled when changes are made
 - eliminates time to repeatedly link executables during development

Appendix A

Lists

Internet Resources

- ☆ • **Sun Java page** (includes API documentation)
www.javasoft.com
 - ☆ • **Gamelan** (contains free source code)
www.gamelan.com
 - ☆ • **Digital Espresso** (summary of newsgroup postings)
www.io.org/~mentor/phpl.cgi?J___Notes.html
 - **Advanced Java mailing list**
advanced-java@xcf.berkeley.edu
 - **Sun tutorial**
www.javasoft.com/tutorial
 - **Elliotte Rusty Harold's FAQ**
sunsite.unc.edu/javafaq/javafaq.html
 - **Search the Sun API and tutorial**
www.kinetiks.com/apisearch.html
 - **How Do I ...**
www.digitalfocus.com/digitalfocus/faq/howdoi.html
 - **Cafe Au Lait**
sunsite.unc.edu/javafaq
 - **Yahoo Java page**
www.yahoo.com/Computers_and_internet/Languages/Java
 - **Java newsgroup**
comp.lang.java
-

Current Licensees

- **Adobe** 12/6/95 - incorporating Java support into its PageMill Web authoring software and Acrobat electronic document software
- **Borland** 11/8/95 - added Java support to their C++ development environment, released a graphical Java debugger, demoed a JIT compiler that is 2 to 12 times faster than the Sun Java interpreter, creating a new Java development environment called Latte
- **Dimension X** 5/1/96 - Liquid Motion web development tool
- **IBM** 12/6/95 - ported to OS/2 and AIX, working on Windows 3.1 port, incorporating into Lotus Notes, adding Java compiler to VisualAge development tool
- **Macromedia** 10/30/95 - incorporating Java into a web authoring tool
- **Metrowerks** 11/10/95 - adding support for Java to its CodeWarrior C++ development environment
- **Microsoft** 3/12/96 - incorporating into its Internet Explorer web browser
- **Netscape** 5/23/95 - incorporated into web browser
- **Novell** - 3/21/96 - allowing Java interaction with NetWare and rewriting many of their tools in Java for portability

Current Licensees (Cont'd)

- **Oracle** 10/30/95 - incorporating Java into PowerBrowser application framework for accessing applications and documents through networks, applet generator in Sedona OO tool suite will allow Java applications to access Oracle databases, Designer 2000 modeling tool will generate Java applications
- **Pierian Spring Software** 5/7/96 - publisher of interactive courseware titles and multimedia publishing products
- **Santa Cruz Operation** 4/30/96 - incorporating into the SCO UNIX operating system.
- **Silicon Graphics** - creating Cosmo Code development environment
- **Spyglass** 11/8/95 - adding Java support to its web browser (Spyglass is a spin-off of NCSA)
- **Symantec** 12/13/95 - adding support for Java to its C++ development environments (Espresso and Caffeine) and creating Cafe development environment

Appendix B

Detailed History

History of Java

- Early '90 - Sun programmer Patrick Naughton threatens to leave Sun for Next, Sun chief Scott McNealy asks for solutions to improving Sun
- Spring '90 - Naughton, James Gosling (author of Berkeley UNIX, UNIX emacs, and the NeWS window system; wrote original Java compiler in C), and Mike Sheridan develop idea to build an environment for “simple” computers
- Spring '91 - “Green team” starts work on a prototype to control consumer appliances
- August '91 - “Green team” wants to build an interface to cyberspace, Gosling creates OO language Oak after abandoning C++ for lack of reliability, Naughton creates animations for interface
- August '92 - demo using “Duke” to control a VCR shown to McNealy
- October '92 - Sun creates subsidiary FirstPerson to develop and market Oak
- Late '92 - Attempts to market phone and TV oak-based interfaces to Mitsubishi Electric and France Telecom fall through

History of Java (Cont'd)

- Early '93 - Arthur van Hoff and Sami Shaio join FirstPerson
- March '93 - FirstPerson loses bid to provide interactive TV set-top boxes to Time Warner who chose SGI instead, SGI fails to meet cost target, Jim Clark of SGI later leaves to found Netscape
- June '93 - Marc Andreessen and Eric Bina, at U. of Illinois National Center for Supercomputing Applications (NCSA), release Mosaic
- Summer '93 - FirstPerson fails to make deal with 3DO
- Spring '94 - FirstPerson dissolves
- Summer '94 - Sun renowned programmer & co-founder Bill Joy and Sun chief technology officer Eric Schmidt gets team to adapt Oak for internet use, Gosling adapts Oak, Jonathan Payne and Naughton write app. to showcase it called WebRunner
- December '94 - Oak and WebRunner were made available to a select group on the internet, Andreessen, who had left NCSA for Netscape voiced his excitement in the San Jose Mercury News

History of Java (Cont'd)

- January '95 - Oak renamed Java, WebRunner renamed HotJava, Naughton leaves to start Starwave which creates online services using Java
- Spring '95 - Arthur van Hoff rewrites javac compiler in Java (new platforms now only need to port the Java interpreter), Sami Shaio writes the GUI library called AWT
- May '95 - official announcement of Java is made, several alpha and beta versions follow
- August '95 - Netscape becomes first Java customer (rumored to have paid only \$750,000)
- January 9, '96 - Sun creates JavaSoft unit
- January 23, '96 - Sun releases Java 1.0
- January 28, '96 - van Hoff, Shaio, & Polese from JavaSoft and Payne from Starwave announce they are leaving to begin a new company