

[IBM](#) : [Developer](#) : [Java™ overview](#) : [Library - papers](#)

Java: Common-Sense Computing For Today's Networked Companies

Mike Quigley
Program Manager, Java Industry Solutions, IBM
August, 1997

This paper is also available in: [Hindi](#) and [Russian](#)

Abstract

The traditional Information Technology (IT) trend toward proprietary technologies that add to the costs and the implementation headaches of IT solutions without delivering additional value must stop. Only when we stop these practices will customers recognize the true benefits that modern technology is capable of delivering. Vendors should compete on their ability to apply the technology to business problems, not on their ability to put the most proprietary hooks into their products or "platforms." With Java, we have an opportunity to concentrate our resources on developing the portable, reusable, and robust environment that we have sought for years. It is time to put marketing discussions aside and let common sense reign.

Information Technology and Your Business

The continual evolution of the business environment has created the need to do far more with much less. Success, once something that had significance over time, is now fleeting. You are only as good as what you deliver today.

Every business is continually seeking tools that can help it meet its objectives. Over the last few decades, computers and networking have been viewed as the quickest way to gain and maintain an advantage. Computers are a tool to help improve business results. As long as they deliver the functions and the return on investment that they promise, they are an asset. When they don't deliver on these promises, they are a liability.

In the past, companies in the computer industry have focused on differentiating themselves through the addition of proprietary technology. This approach has produced a myriad of bells and whistles and some very "cool" functions that thrill computer whizzes; it has also created rival factions within companies that argue over which widget is best. This has probably contributed as much as anything to diverting our focus from how technology can help our businesses. It has added both costs and inefficiencies to virtually every phase of solution development and deployment. What it hasn't produced is a technology environment that enables you to receive the best return on investment in information technology.

Billions of dollars and hours are spent every year to integrate disparate products or build solutions. The problem is that much of this money is wasted writing customized "glue code," integrating different vendors operating systems, or porting applications. If we are ever to realize the maximum return on our investment, we must element the inefficiencies that the industry has created by focusing our efforts on the development of many different proprietary "platforms." We have learned how to make our businesses far more flexible and

efficient by eliminating unnecessary costs. Now it is time to do the same to information technology.

This paper is focused on looking at the relationship between the business and information technology and asserts that we have a rare opportunity to make a significant leap in improving this relationship and stopping the waste of billions of dollars and programming hours. We can eliminate many of our day-to-day problems in this industry by joining forces and focusing on the development and maturation of Java. We must make it the de facto standard, maintain its ubiquity, and have it become an enabler for all companies, rather than a differentiator for one or two companies.

Business Today

Today's business model is a successful hybrid of several preceding models. It combines the focus on integration, reliability, and availability of the traditional, centralized company with the flexibility and speed of companies organized more along departmental lines. Most of all, it recognizes the importance of speed to market, flexibility, cost efficiency, and focus on core competencies.

Modern products are designed, built, marketed, and sold by teams. These teams often form "virtual companies" that span internal departments and external suppliers. Businesses recognize that these people must all work together as a cohesive unit in order to achieve business objectives, and that accurate and timely communication is paramount. Anything that distracts from their focus on the business task, inhibits them from reacting quickly to changes in the market or adds unnecessary costs is avoided.

We have examples throughout our daily lives. Your home is probably unique in many ways, but, because the raw materials that went into it were mostly built to a common set of standards, you are able to purchase it at a value. How much would your home have cost if every lumber and accessory supplier differentiated their product through dimension, not quality or other means? What would a lamp cost to purchase and install if every manufacturer of electrical outlets had a unique plug configuration? Dramatically increased costs with no increase in value: not a good equation. This isn't rocket science; it is basic business know-how and common sense. Isn't it about time we applied that same logic to the information technology industry?

Information Technology Today

The approach that we have taken in information technology has been much different. The IT industry has focused on differentiating our companies through the development of proprietary technologies. Companies have tended to specialize in "marketecture" and hype. Companies espouse their superiority in the "platform" business. Stop and think about it for a minute. The technology that enables business applications to talk to the hardware and network is a necessity, but the more types we have the more unnecessary cost and complexity is added.

This approach has added to the costs, development cycle times, and implementation headaches of IT solutions without delivering any additional value. Why, then, have we let ourselves get so focused on spending billions of dollars to build so many different operating systems that we then must spend billions more so that we can use them in our business. Does this make any sense to you?

Today's companies face many different considerations when selecting hardware and software to run their business. They must weigh what hardware their end users have, what operating system is preferred by certain groups, what upgrades are necessitated by their decision, how the new products will be deployed, or what the future support costs will be. Vendors produce business cases showing how products will improve individual productivity or reduce costs. They compete on their ability to deliver the latest and greatest technology to business problems while justifying the right to have proprietary products.

The Common Sense Answer

The answer is simple: Stop pouring money and resources into the development of proprietary technologies and focus them on building a common technology infrastructure that we all then leverage to build the business applications of the 21st century. Common sense tells us that, if we take only a fraction of the resource we spend on developing proprietary base operating systems and middle ware and worked towards building this standard, we could easily address all of our requirements, save billions of dollars and programmer hours, and

become more competitive in the marketplace. Virtually every industry has recognized this and gone to this model. Isn't it about time the computer industry did as well? The opportunity is at hand with Java.

We should stop trying to see who can build the most unique and proprietary features into our products and solutions. It only costs our customers time, money, and lost productivity. We should focus our efforts on making Java successful. It might hurt the feelings of some who thrive on controlling customers through the use of proprietary technology, but they'll just have to get over it.

How did we get where we are today?

Information technology has come a long way in a very short period of time. In order to better understand the premise of this paper it is important to take a brief look at the evolution of the industry. It is important that we understand the path that we took to get to the model we are on today, so that we can learn from our past. Let's leverage the good decisions and avoid repeating the bad ones.

The Early Days of Business Computing

Information is an important asset to any company. Often, companies succeed or fail based on how they gather, evaluate, and apply information. With the dawn of the computer age, businesses began to look to computers to help them improve. Early systems focused on reducing paperwork and "automating" redundant processes. Companies were vertically integrated and wanted to own and control every phase of the development, production, and marketing of products. Since most companies were very hierarchical and centralized, the computing model followed suit. Employees accessed data and applications on centralized computers. As data (and the applications that gathered, stored, and manipulated the data) grew, so did the mainframe environment that supported them

These systems evolved to be reliable, available, and dependable; however, they had their drawbacks. The technology tended to be very inflexible and difficult to change. Two- to three-year development cycles for new applications were the norm, with many companies having years of backlogs in application-development groups. Change processes, while efficient, were long and arduous. At the same time the marketplace, and therefore the business model, needed to be successful and began to change. The technology couldn't keep up with the changes in business.

New concepts included autonomous business units, leveraging business partners and suppliers, and a real focus on reduction of cycle times. Business units within a company were often asked to compete against one another or outside firms for business that had once been theirs by default. At the same time, innovations in IT began to occur at an increasing pace. New technologies offered more flexibility, much shorter development times, and freedom from centralized IT. PCs and mid-range computers brought with them a tidal wave of new personal productivity and business applications. This paradigm shift away from a centralized IT structure and towards departmental computing often delivered function to the individual user or department but not necessarily to the corporation as a whole.

The Decadent Desktop

The paradigm shift towards autonomy accelerated and the personal computer promised improved personal productivity and local applications. Business unit budgets, once dedicated to supporting the centralized IT shop, were now empowered to buy new hardware and killer departmental applications. Client-server computing became the cool alternative to those nasty old mainframes. Development times were months, not years, and the information was on your desktop. Why pay millions for computing power when a thousand or so would buy a workstation?

The frenzy increased as prices of these units, extra memory, larger hard drives, and other accessories began to drop dramatically. Workstations became so popular that many businesses focused on deploying one on every desk, regardless of the usage characteristics of the employee.

Another frustrating factor is the very short life cycle for PCs. How many of us spent a few thousand dollars to buy the latest and greatest PC only to see a cheaper, more powerful model within a couple of months? For your personal use this is discomfoting, for a business it is disastrous, and not just from the financial aspects.

New technology often drives ideas for new applications. Applications are often selected by "power users," not the much larger number of employees that are occasional users of the application. How many employees in your company really leverage all of the power of applications that are available to them? Not only are we spending dollars for unused function, this environment can create a "snowball" effect where new application function drives greater hardware requirements. The cycle repeats itself every few months. How much time and money does your company spend distributing new versions of PC software? How many different versions of any given product must you support at one time? Is your company getting a good return on this investment? How can a company keep up fiscally or operationally?

Sometime in the early 90s we began to look seriously at the whole relationship between the IT and business model. Consultants began producing volumes of reports that established the cost of maintaining a single PC at anywhere from \$8,000 to \$15,000 per year. Vendors introduced PC hardware "upgrade" programs and software vendors went to 6- or even 3-month product release cycles. Nice from a product function perspective, but how fast can your company absorb changes?

During this period we totally lost focus on why we turned to computers in the first place. Instead of focusing on how they contributed to the business, we became enamored by the technology. Hardware and software rolled in the door in waves. Each new wave brought with it lots of new features and functions, but were the benefits really there? How many of us use all of the IT function that is available to us today? How much money do businesses around the world spend buying, implementing, and supporting technology that isn't used? Is the technology really contributing the maximum to bottom line profits?

Attempts to change

While standardization is a common sense approach, the computer industry has not been able to enjoy the potential benefits of a standards-based approach. We have tried to standardize data models, network protocols, and operating systems. Today, there are over 1500 "standards" bodies in the IT industry. Manufacturers, while publicly endorsing standards, has felt the need to differentiate their implementation of the standard by delivering a deviation that included their "added value." Instead of differentiating product lines, this "maketecture" approach has created a world of proprietary systems that don't talk to each other, require significant investment to integrate and support, and tend to force the customer into a path almost totally controlled by the vendor. This also forces companies to spend a lot of time and money that provides no return on their investment.

Network computing

We are on the brink of a new era in information technology. We have talked for years about the benefits of portable applications, code reuse, and rapid application development. Now it is time to deliver.

Delivering on the Promise of Network Computing

We have learned a great deal from these experiences. We are moving at unbelievable speed to take what we have learned and apply it to a new "networked computing" model. Today's businesses deserve, and are demanding, a computing environment that leverages what we have learned and delivers on these promises. The industry must eliminate these inefficiencies and proprietary approaches and truly improve the return on their investments

The computing environment of the 21st millennium must be one where vendors and companies are not encumbered by the worries of how to handle differences in technology. The environment must be one in which vendors and companies differentiate themselves not by their ability to create proprietary environments and cover them up with great marketing messages, but by their ability to apply knowledge and intellectual property to business problems.

The Benefits of Java

Java offers us the opportunity to do just this. Java is not the answer to every problem, but it is a solution to one of our biggest fundamental problems and can help us establish a new method of working as an industry. Today, most of the industry is rallying behind Java. Companies are contributing millions of dollars worth of

intellectual property to the Java Developers Kit and the Java Virtual Machine-property that once would have been used to build a proprietary competitive edge. Think of the power of this. Tens of thousands of developers that were once doing duplicate development and figuring out proprietary ways to implement a single function, are now working on a common code base. National Language Support, a necessary but difficult area, is now available to everyone. Small companies now are building applications that are available to the world. This enables the technology to be made available to everyone and greatly reduces the cost associated with developing new applications or capabilities. Development resources once dedicated to porting code now are freed-up to build applications.

Vendor Benefits

Software development projects historically have incorporated millions of dollars and development resources dedicated not to writing logic to improve the application, but to porting testing, and maintaining the application on multiple proprietary platforms. These tasks are not fun for developers and often delay getting needed function to the customer. The availability of the Java Virtual Machine on every major hardware and software platform means that we no longer have to budget dollars for porting costs or tie up highly skilled programmers in the boring and frustrating task of porting code or supporting numerous, operating system-specific versions of an application.

It is also important to differentiate between the traditional, proprietary approach and the Java approach. The traditional approach requires supporting a myriad of platform specific APIs. Java and the "write once, run anywhere" approach requires supporting a single set of APIs that run on any platform.

Vendors are now free to focus on applying these skilled resources to delivering more and better applications. The cost of building and supporting applications go down. These are all real, tangible benefits to the vendor community.

Customer Benefits

Customers either spend a great deal of money on integration projects or have to defer implementation of business decisions because the integration costs are prohibitive. Rollout and support of new function requires tedious planning. In-house development also faces the same challenges as the vendor community. With Java, most of these problems go away. More function that is easier to implement means improved return on information technology investments. A level playing field for vendors and their reduced costs means more competitive application prices. Again, tangible benefits for the customer.

Benefit of JavaBeans

One of the promises of the past was code reusability; Java delivers with JavaBeans. Need a function to track name and address? Build it as a JavaBean. After it's built, it can be shared, sold, or reused. The whole idea of JavaBeans promises to deliver vast reusable libraries of functions that can quickly be easily assembled into new applications to meet urgent business requirements. The JavaBeans specification means that now AD tools, by adapting to the specification, can freely interchange beans, thereby greatly increasing your flexibility to choose the AD tool that best suits your needs. And you won't have to modify the code to run on the 27 different hardware platforms our company has installed. Talk about flexibility.

The JavaBean specification is also an excellent example of what we can achieve if we quit fighting over technology and cooperate. The specification was developed by 14 companies in less than 60 days. See what the industry can deliver when it wants to!!

Write Once, Run Anywhere

Portability and flexibility, both information technology promises of the past, are a reality with Java. In the past, we have had languages that espoused portability, but none delivered anything remotely resembling portability. Developed from the ground up as object oriented and portable, it has already addressed the biggest challenge. Companies now are free to let departments select the hardware platform of their choice without worrying or having to budget for integration and porting costs.

Java applications are implementation-neutral. We often get caught up in the fat client, thin client rhetoric.

Because Java is truly object-oriented, you are free to distribute the application business logic in the manner that best fits your needs. Modifications are made easily and transparently as business needs change.

Rapid Application Development

JavaBeans, true object-orientation, and the ability of AD tools to easily inter-operate deliver on the value and promise of rapid application development. Java projects at companies throughout the world are delivering real business value in weeks and months, not years. Companies are able to leverage existing systems and data, enter new markets, and even develop new businesses in record times.

Speeding Ahead

Let's put all of the marketing hype and focused messages aside. Let's take this opportunity to really exploit the potential offered by information technology. Some are threatened by this potential and are quick to point out the shortcomings of Java today. Is it perfect? No, but then the nay-sayers probably have a few holes, security problems, or glitches in their systems, too. What Java is doing is maturing at a rate that is unprecedented in the industry. Just as businesses have made dramatic reduction in cycle times through cooperation and standardization, the information technology industry is changing the way we do business by cooperation on the development of Java. We can accelerate this pace even more we stop spending time and money on promoting what it won't do and focus those resources on improving it. A small book of excerpts from John M. Capozzi's book *If You Want the Rainbow You Gotta Put Up With the Rain* contains the following quote: "Nothing is more rewarding than to watch someone who says it can't be done get interrupted by someone actually doing it." It is time that we as an industry stop talking about why it can't be done and set about the task of doing it.

Act now

Companies throughout the world are beginning to implement Java solutions, but it isn't enough. We need to push it, stretch it, and even break it. If we don't, it won't change fast enough. We need to support vendors and customers that add to the technology and adhere to the requirements in order to keep Java truly open, ubiquitous, and portable. 100% Pure Java shouldn't be viewed as a marketing campaign; it should be viewed as the logical, common-sense approach, the only approach that will truly move the industry forward and unleash the full potential that has been held back for so long.

Some in the industry have criticized this initiative but, again, this is just the ancient "marketecture" approach designed to protect the proprietary environments and eliminate a customer's freedom. The goal of the campaign, and the benefit, is to make sure that applications are truly "Write once, run anywhere." We have to stop getting distracted by individuals and companies that continually tell us that we shouldn't listen to common sense.

This is not the time to sit on the sidelines and wait to see if Java is going to succeed or not. We have an unprecedented chance to take a shot at revolutionizing the industry. The only way that we will reduce costs, improve our return on investment, and truly exploit the potential of information technology is to get involved. Contribute, develop, or evaluate, but above all do not accept anything that furthers the proprietary and wasteful ways of the past. Hockey legend Wayne Gretzky once said, "You miss 100% of the shots you don't take." Take the shot; it can't miss.