The Computer Systems Innovation Group

Implementing Java

By Sergio C. Carbone

Welcome to Implementing Java

- Introduction
- How you can start using Java now
- A case example: Searching at Softwrite
- What you need to get started
- Conclusion

Introduction

- We have all heard about Java and seen technical information about its capabilities, but when it comes time to work with it, many people hesitate.
- The real value companies have in computer systems is found in their existing systems and knowledge base.
- A company’s systems provide valuable services, and developers know how to leverage their existing tools to deliver results.
- Building or rewriting a system completely in Java means loosing the real value of an organization. It means starting from scratch.
- Complete Java implementations mean high exposure, because all aspects of system development change and all the ‘pit falls’ need to be learned again.
- These points make developing solutions completely in Java unattractive.

What Is Needed

- A realistic approach to Java development is needed.
- An approach that blends Java with a company’s existing investment in databases, file sets, job streams, and processes.
- A way of acclimating developers to a new development environment with control, so their business and technical knowledge is leveraged with Java.
- A guarantee that valuable working systems will not be thrown away to jump into a black Java hole of risk.
- An approach that will use what we have to get what we want, but will still be founded on solid systems principles and support real object oriented Java.
- The way exists and it is available to everyone.

How You Can Start Using Java Now

- There is a realistic approach to Java development.
- It will extend our existing investment in databases, file sets, job streams, and processes. It will provide a controlled acclimation of Java to developers.
- It will add to our business and technical knowledge, work with valuable existing systems, and be founded on solid systems principles that support real object oriented Java development.
- The approach involves using our existing systems as large grain objects, and accessing them via JDBC and DB2; this is applicable to all AS/400 systems, even if they do not use DB2 directly.

Benefits of Leveraging Existing Systems

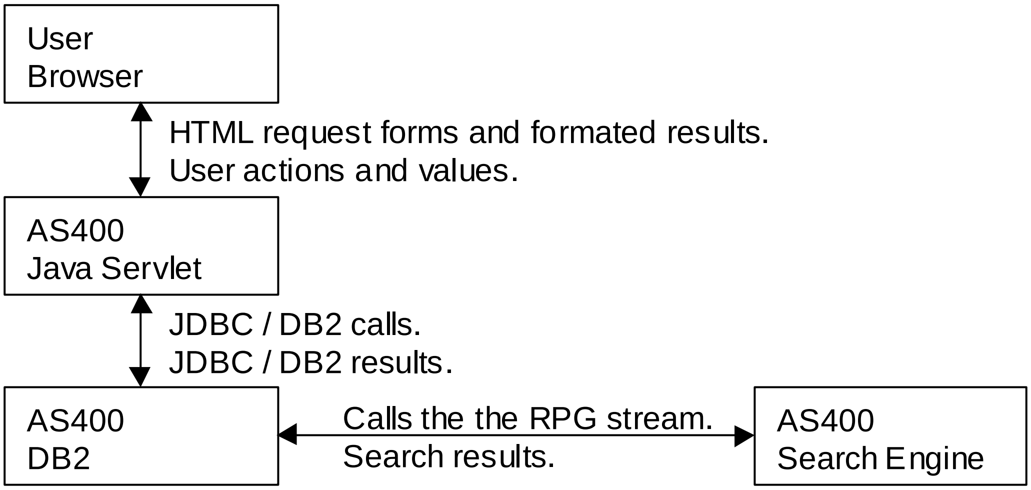
- By building onto proven applications, risk is isolated and reduced, and projects have better focus and higher success rates.
- Making use of existing systems to do the bulk of the processing will offer the highest performance. In this way we use tools for what they are best at.
- Because the amount of Java that will be developed is reduced and well focused, developers can be exposed to the Java environment in small teams with high odds for success.
- By integrating Java with existing systems, developers who know the existing systems can overlap both environments, so there is less resource down time and projects can keep moving.
- There will be some modifications that need to be made to the existing systems so they can be used as large grain objects. However, experience has proven these modifications of existing job streams to be quick and seamless.
- This is the fastest approach to delivering value with Java while keeping with sound development and object oriented principles.

A Case Example: Searching At Softwrite

- Upon meeting with Softwrite Computer Systems, we established a goal of building Java expertise and ultimately exercising that expertise in converting a highly valuable existing system to Java.
- A plan was established that involved Java and object oriented education, mentoring, architectural planning, and guided development.
- The system that was selected for conversion was a proprietary search engine.
- Historically, the search engine was known for its high speed and flexible searching capabilities. Any new version would need to deliver similar results.

The Solution

- After reviewing several different options, the architectural plan that was implemented blended a new Java based servlet with a slightly modified version of the existing AS/400 based search engine.
- The new approach specified that the Java servlet would generate the HTML user interface, and pass user requests onto the AS/400 search engine by calling through JDBC / DB2.



- The results would be returned to the servlet via JDBC / DB2, and the servlet would generate the HTML that the user would view through any browser.

The Solution Benefits

- This approach offers the highest performance to the user, and minimizes client side resource demands.
- It produces low network traffic; users are passed results in HTML documents.
- Java, in this case, was used as a vehicle to disseminate information and provide access to an existing system.
- The modifications required to the existing system were minimal, and the biggest change was developing a module that was available to DB2.
- Almost 90% of the system's functionality is derived from existing code.
- With this effort, a system that was bound to the AS/400 is available to be leveraged by any platform, and offers high value for a low cost.

What You Need To Get Started

- A professional Java development environment with integrated debugging.
- AS/400 WEB server services and DB2 with the DB2 JDBC classes.
- Formal training in Java and object oriented concepts; Java and OOD are not conducive to self-instruction.
- An understanding of how to leverage architectures and systems.
- A mentor or supporting group that is experienced in Java and OOD.
- Determination and commitment to using Java and a focus on delivering real business value.

Conclusion

- The way to deliver real value is by enhancing what is currently delivering value, and leveraging the knowledge of your systems group.
- Java does not need to be all or nothing, and Java solutions can be built to offer the best performance and lower resource demands.
- The key is choosing the correct direction and having the perseverance to follow it. With this, everyone will be successful implementing Java.
- Try… Try… Try again…
- Thank you for listening to my ideas, have a great time with Java, and if you have any questions, please contact me.