Java World
Fueling Innovation ™

January 2002 | HOME | FEATURED TUTORIALS | COLUMNS | NEWS & REVIEWS | FORUM | JW RESOURCES | ABOUT JW

A Java case study: The power of J2EE

How the Energy Information Administration secured a complete J2EE solution in less than five months

Summary
In "Step into the J2EE Architecture and Process," Jian Zhong introduced an eight-step cycle methodology for Java 2 Platform, Enterprise Edition (J2EE) development. Here, he and Betty Barlow of the Energy Information Administration (EIA), United States Department of Energy, present a real-world case study that uses that simple methodology to build an e-government solution for EIA. Discover how, in the agency's first J2EE effort, many benefits were gained by using J2EE; specifically, learn the advantages of the J2EE security model and how application security was accomplished. (*2,500 words;* **January 18, 2002**)

By Jian Zhong and Betty Barlow

The Energy Information Administration (EIA) collects data from the energy industry on nearly 80 survey instruments, managed by several program offices. In 1996, realizing all these surveys shared many similarities, the Business Re-engineering Steering Committee tasked the Office of Information Technology (OIT) to build a multimillion-dollar, one-size-fits-all collection and processing system using Microsoft technologies. Satisfying every program office's specific requirements proved difficult with the technology available at the project's outset. However, due to the Internet's rapid growth over the last few years, many EIA customers preferred to submit data electronically. (Federal law will eventually require government agencies to provide an electronic option.) In view of these developments, EIA decided to focus on developing an Internet data collection application.

Betty Barlow was assigned as the project manager for EIA's Internet data collection project. She managed development activities, such as project planning, requirements analysis, design and implementation, testing, deployment, and so on. She also coordinated the development team's activities with a representative from the program office, for whom we were doing the development.

ActioNet was brought in to support the EIA CIO's vision of a multitiered enterprise architecture for EIA-wide survey applications. Our first project was the EIA-877, the State Heating Oil and Propane Program (SHOPP), a survey managed by the Office of Oil and Gas (OO&G), Petroleum Division. Jian Zhong was hired in mid-May 2001 to help EIA jumpstart the project. Jian's major contributions were to advise less-experienced OIT personnel working on the SHOPP project on multitiered applications, help to establish an enterprise and application architecture, build development and production environments, and ultimately deliver a complete e-government solution -- this time using Java 2 Platform, Enterprise Edition (J2EE) technologies -- by October

1, 2001. The deadline was tight because the EIA-877, a survey of winter fuels' distributors, starts on October 1. The risks were high because ActioNet's future at EIA was based on its performance in building EIA's first Internet data collection instrument. Even though Jian was hired as a senior software architect, he was the only Java developer on the project until September when EIA hired a second one.

High-level project requirements

The respondents, who are state energy offices, call companies assigned to them to get price information for propane and heating oil during the winter heating season. EIA determines the list of companies assigned to a state and the product list for each company at the season's start. In prior heating seasons, the information collected was transmitted to EIA electronically via Petroleum Electronic Data Reporting Option (PEDRO). This system was a first effort to give the respondents a vehicle (sent to them on diskettes and loaded onto their PCs) for entering their data and sending it electronically to EIA via modem or by using nonelectronic means either twice a month or weekly.

EIA sent the PEDRO software to respondents at the beginning of the season. The respondents entered the data into the application and transmitted it to EIA via modem. If EIA changed the company slate or the product mix during the season, it notified state energy office personnel, who then had to enter the changes in the application. To replace this labor-intensive process, OO&G wanted a system that would let respondents input and submit their data via the Internet, which would allow OO&G to control the company/product mix and send notices to respondents. Figure 1 shows a high-level use case diagram for the survey.
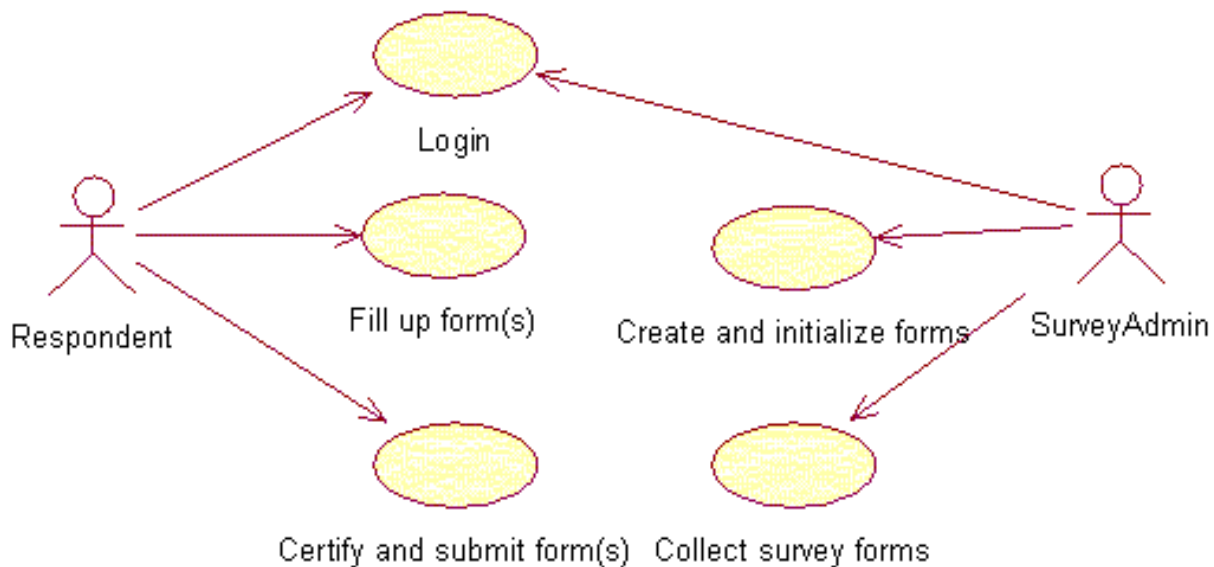

Figure 1. Survey use cases

How we introduced Java

As in many large organizations, EIA has several different operating systems running many kinds of software systems. The primary platform for business critical data is the IBM mainframe S/390. EIA also has a major investment in Oracle 8i on the S/390. Of the hundreds of legacy applications running on the mainframe, many are quite a few years old.

The advantages of using the IBM S/390 include its reliability, automated backup, scalability, flexibility, and

security. The computer S/390 and operating system OS/390 have been running continuously for five years in EIA without significant problems. The tape robot and the automatic system on the S/390 provide a completely automated, fast, and reliable file backup and restore system.

We can partition the S/390 to run many independent systems simultaneously. We can easily add hardware resources, such as CPU, memory, disk, or network adapter, to the S/390 to increase capacity without making any change to the operating system or application systems. In addition, we can redistribute the CPU, memory, and disk based on the ever-changing application requirements. The Access Control Facility, type 2 (ACF2), the method EIA uses to control access to the OS/390, is known for its tight and reliable security; thus far, there are no known instances of hacking into the OS/390.

Traditionally, there has been a downside to the mainframe. The typical user interface for the S/390, the 3270 terminal, for example, is not user friendly. Prior to the commencement of EIA's Internet data collection project, another group spent a few months proving that it could develop Java on Windows and deploy it on the mainframe. This solution solved the interface problem and made the mainframe a viable platform for Web-based applications. Java's platform independence allowed us to take advantage of both the mainframe for deployment and Windows for development.

Build enterprise architecture with existing infrastructure

As discussed in "Step into the J2EE Architecture and Process," the enterprise architecture reflects an organization's long-term goals. It is the base upon which the application architecture is built, but you cannot upgrade it overnight. When budget is a limiting factor and in an economic downturn where every developer is competing for business, it is critical to deliver a solution quickly with an organization's existing infrastructure. One amazing thing about the SHOPP project is that we didn't buy any new hardware and only one piece of software. The IBM HTTP server, WebSphere application container, and Oracle 8i Enterprise RDBMS (relational database management system) all pre-existed on the legacy IBM mainframe enterprise server. We used Microsoft Visual SourceSafe (VSS) for source code version control because OIT already owned it. The only software we bought was an IDE -- JBuilder 5 Enterprise. Figure 2 illustrates the enterprise architecture we built as a UML sequence diagram.
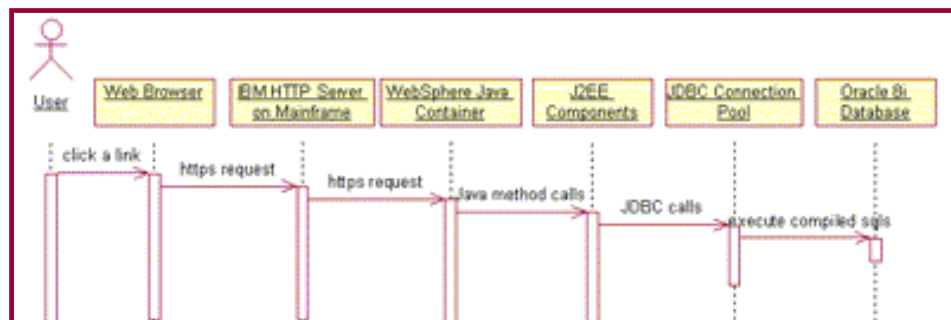

Figure 2. Enterprise architecture sequence diagram. Click on thumbnail to view full-size image.

Software development lifecycle methodology

We can never overemphasize the importance of formal software process and methodology, especially in the government consulting service sector. For any project, usually many different parties are involved, including personnel from different departments, both federal employees and contractors. And the contractors might be from several competing companies. One of our requirements was to follow the Department of Energy's Software Engineering Methodology (SEM). We are lucky that SEM resembles Jian's J2EE methodology introduced in his last article. SEM is an iterative and incremental lifecycle methodology whose major stages

are planning, requirements definition, architectural design, detailed design, programming, testing, installation and acceptance, and software maintenance.

Use J2EE BluePrints as application architecture baseline

Before the project began, not many people at EIA realized the importance of an enterprise application architecture. The proof was that we could use an existing hardware and software infrastructure and a small set of already-contracted developers to deliver an Internet data collection project on time with limited resources. Jian spent a significant portion of his time in the first few months participating in discussions, such as what is Java, what is byte code, how to compile Java source code, how to use a package, how to do source code control and why it is needed, how to develop a multitiered Internet application, and why an application client is not allowed to access a database directly. We attended endless meetings and had to justify many obvious architectural decisions in case they conflicted with previous practices.

J2EE is one of the best server-side technologies, and *Java BluePrints* for the enterprise can and should serve as important guidelines for the first few projects. We selected J2EE technology because it offers so much at the architecture level, including Enterprise JavaBeans (EJBs), JavaServer Pages (JSPs), servlets, Java Message Service (JMS), transactions, CORBA, JDBC (Java Database Connectivity), XML, JNDI (Java Naming and Directory Interface), and so on. Our SHOPP application includes the following architecturally significant components:

1. Browser clients: Respondents use this graphical user interface (GUI) to interact with the Internet data collection system

2. HTTP server: IBM's HTTP server, a variation of Apache, handles initial HTTP requests from users and forwards all requests to a Web application container running on the mainframe

3. Web application container: Serves as the running environment for all Web components running on the mainframe, such as JSPs and servlets

4. JDBC connection pool: Implements application interface between Java Web applications and the Oracle database

5. Database: Oracle 8i database running on the mainframe contains business critical data

We decided not to use applets. Instead, we put most of the business logic into Java libraries and presentation logic into JSPs. Applets would have created a richer user interface, but because potential firewalls between a user's browser and EIA's Web server could cause browser compatibility issues, we decided to use JSPs for our entire dynamic user interface. We also decided not to allow application users direct access to EIA's backend database. It was important to have application users authenticate themselves to the middle tier and have the middle tier authenticate itself to the backend. The respondents (who have access only to their own data) and admin users (who have access to all data) are separated via roles. Fine-grained access control, such as preventing users in the same role from seeing each other's data, is implemented within the application server as dynamic views and an auditing component in addition to database security measures. Figure 3 shows a domain object model about the respondents and survey forms.
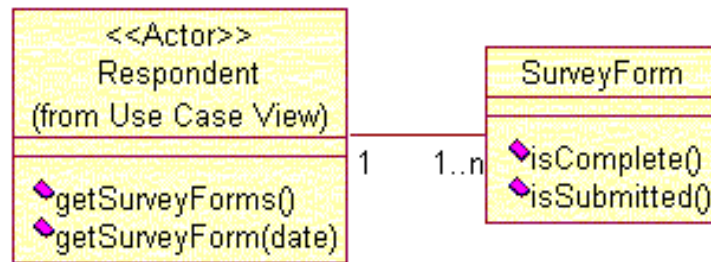
Figure 3. Survey form class diagram

Project staffing -- the players

Many people who would like to adopt the J2EE model often hesitate because of concerns about their current developer skill sets. Our experience at EIA shows that you need only a few experienced Java developers to enable many types of IT shops to become complete J2EE solution providers. One J2EE advantage is the ability to reuse legacy systems as well as reuse legacy skill sets. With limited resources, we delivered a functional system that fully satisfies the users' requirements and adds capabilities unavailable in the previously used application. We evolved a software development process that fully utilizes OIT's existing resources and complies with the SEM.

Many people contributed to SHOPP's success: program managers, GUI designers, database administrators, architects, Java programmers, configuration managers, technical writers, testers, security personnel, communications experts, and mainframe system administrators. Because J2EE uses a highly modular component approach, people with different skill sets could work on different areas concurrently and without much conflict. With our J2EE architecture, creative graphic designers could concentrate on presentation logic while Java programmers focused on computation-intensive business logic. The EIA project was truly a model for how an IT shop's various groups can work together to build an excellent product. We developed a solid, multitiered architecture from scratch, which EIA can use for many other projects.

Establish a security architecture and process

Internet application security is an important factor for any e-government solution. On the one hand, you cannot afford to deliver an application that adds too many risks to the infrastructure or application data. On the other hand, the resources are often limited. There are no absolutely secure Internet application systems in the real world. Tradeoffs between security and convenience/cost always exist. If the cost of breaking into a system is significantly higher than the benefits gained by attacking the system, and the cost of protecting the system is lower than the value of what is being protected, then we call a system secure. When we were asked by senior management "Are you sure your application is 100 percent secure?," our answer was "No." We tried very hard to let the decision makers know the costs and risks of different kinds of Internet systems. First, we embraced J2EE BluePrints as our architecture baseline. Second, our security and application architecture was extensively reviewed by in-house security officers and independently evaluated by third-party security specialists. Finally, high-level management approved the security architecture.

One of the J2EE security model's major advantages is that J2EE defines standards in the specification and its BluePrints offers best security practices without dictating how to implement security. Because of this, we implemented our application security with ease using our existing security infrastructure and process. Another advantage is that you could develop a security architecture for your particular need. For example, our application deals with nonclassified information, thus the security requirements differ from those dealing with top secrets. We cannot imagine anyone inventing one security architecture that satisfies all requirements for all kinds of Web applications.

Results and benefits

Our project's ultimate achievement was a live Web application that satisfies users' requirements. But to give you a feeling of the size and scope of EIA's first J2EE project, here are some details of our deliverables:

1. A set of requirement documentation
2. 51 Java classes with about 6,000 lines of code
3. 43 JSP files with about 12,000 lines of code
4. 25 database tables
5. 28 graphics files
6. 1 configuration file

As SHOPP is EIA's first Internet data collection project, together with building the application, EIA also established the following development and production environments:

1. 1 production system, including HTTP server, servlet container, and Oracle database on the S/390
2. 1 test system, including HTTP server, servlet container, and Oracle database on the S/390
3. 4 development systems, including servlet container, IDE, and database connections to Oracle on the mainframe
4. Source code version control system
5. Bug report and fix system
6. SHOPP application admin user interface for intranet client: Visual Basic/MS Access

Low cost

The last, but most important, point is the low cost of building a J2EE system. As J2EE uses a distributed component architecture, you don't need to rewrite the whole system every time you add a new feature, fix a bug, or just update to a new technology version. The result is very low cost. We developed a live system from scratch in less than five months with essentially three developers. Of course, many other OIT personnel played indispensable roles in implementing the supporting infrastructure and checking security issues.

User feedback

The user feedback for the first survey has been positive. All 25 state energy offices in the program use the application. Several have commented on how easy it is to use, particularly in comparison to the PEDRO system used previously. EIA's Statistics and Methods Group did cognitive testing with a few users to watch as they navigated through the application and to get their comments and suggestions. Their comments were also favorable, concerning both the functionality provided and the navigational ease.

J2EE aims to please

The paperwork elimination requirement challenges many IT organizations that serve the e-government sector. The organization must deliver quickly but still fully satisfy the customer's requirements. Our experience at EIA has shown us that it is not always easy to develop a one-size-fits-all generic solution for multiple customers, since it is difficult to please everyone. J2EE, as a distributed component architecture, is the right solution in many of these situations. As much as possible, you reuse commercially available or homegrown components in building a system that fully satisfies each customer's requirements. And, with J2EE, you can integrate your new system with legacy systems and save resources.

Jian and Betty would like to thank Mike Lehr, a senior information technology specialist at EIA, for help in reviewing this article.

About the author

Jian Zhong, senior software architect with ActioNet Inc., has been actively involved in J2EE architecture development for the last three years. He started Java programming in early 1996 and has been a Sun Certified Architect for the Java 2 Platform since October 1999 and a Sun Certified Enterprise Architect for the J2EE Platform since November 2000.

Betty Barlow is the Internet data collection project manager and acting division director for the Systems and Policy Division, Office of Information Technology, Energy Information Administration, U.S. Department of Energy.

Resources

- *Java BluePrints:* Enterprise BluePrints, includes design patterns catalog and the pet store demo architecture overview:
  http://java.sun.com/blueprints/enterprise/index.html
- *Designing Enterprise Applications with the Java 2 Platform, Enterprise Edition,* Nicholas Kassem, et al. (Addison-Wesley, June 2000; ISBN: 0201702070):
  http://www.amazon.com/exec/obidos/ASIN/0201702770/javaworld
- *Java 2 Platform, Enterprise Edition: Platform and Component Specifications,* Bill Shannon, et al. (Addison-Wesley, May 2000; ISBN: 0201704560):
  http://www.amazon.com/exec/obidos/ASIN/0201704560/javaworld
- "Step into the J2EE Architecture and Process," Jian Zhong (*JavaWorld*, September 2001):
  http://www.javaworld.com/javaworld/jw-09-2001/jw-0928-rup.html
- Also check out the 23 "Gang of Four" design patterns in *Design Patterns: Elements of Reusable Object-Oriented Software,* Erich Gamma, et al. (Addison-Wesley, January 1995; ISBN: 0201633612):
  http://www.amazon.com/exec/obidos/ASIN/0201633612/javaworld/
- *UML Toolkit,* Hans-Erik Eriksson and Magnus Penker (John Wiley & Sons, October 1997; ISBN: 0471191612):
  http://www.amazon.com/exec/obidos/ASIN/0471191612/javaworld
- *Design Patterns and Contracts,* Jean-Marc Jezequel, Michel Train, and Christine Mingins (Addison-Wesley, October 1999; ISBN: 0201309599):
  http://www.amazon.com/exec/obidos/ASIN/0201309599/javaworld
- Browse *JavaWorld*'s **Java 2 Platform, Enterprise Edition (J2EE)** Index:
  http://www.javaworld.com/channel_content/jw-j2ee-index.shtml
- Subscribe to *JavaWorld*'s free *Enterprise Java* email newsletter:
  http://www.javaworld.com/subscribe
- Speak out in *JavaWorld*'s **Enterprise Java** discussion:
  http://forums.idg.net/webx?50@@.ee6b80a
- You'll find a wealth of IT-related articles from our sister publications at IDG.net

A Java case study: The power of J2EE