Amin Ahmad

http://www.ahmadsoft.org   amin@ahmadsoft.org

Certifications and Training

- IBM Certified Solution Developer – XML and Related Technologies.
- Sun Certified Web Component Developer for Java 2 Platform, Enterprise Edition (310-080).
- Sun Certified Programmer for the Java 2 Platform (310-025).
- IBM MQSeries training.

Publications

- Ahmad, A. (2008, Feb 12). Ropes: Theory and Practice. *IBM developerWorks*.

Education

Bowling Green State University
Bachelor of Science, majoring in Computer Science and in Pure Mathematics
- Graduated summa cum laude (cumulative grade point average 3.95/4.00)

Experience Summary

- Ten years of experience as a senior consultant implementing Java-based solutions for clients in the *financial and software industries* and for *Federal and State Governments*.
- Extensive experience designing and implementing Eclipse RCP applications (utilizing OSGi), as well as writing plug-ins for the Eclipse platform. Extensive experience with the Eclipse Modeling Framework (EMF) and the Graphical Editor Framework (GEF). RCP development for clients including *IBM Research*, *NYSE*, *American Express*, *Allied Semiconductor*, and *eCredit, Inc.*
- Excellent industry experience developing portable web-based user interfaces using the Google Web Toolkit (GWT), AJAX, JavaScript, and CSS 2. Some SVG experience.
- Extensive Java EE design and implementation experience using Servlets, Java Server Pages, Enterprise Java Beans, JMS, and Applets. Experienced in designing for very large systems handling millions of transactions per day.
- Experience with JDBC 2, SQL, Hibernate, and iBatis. Experience with DB2, UDB, Oracle, Firebird, and MySQL.
- Experience with UML design using Rational Rose and Omondo UML.
- Strong background in rich client development using Swing and SWT/JFace toolkits.
- Strong experience utilizing various XML standards and technologies in J2EE solutions, including DTD, W3C Schema, DOM 2, DOM 3, SAX, XPath, XSL-FOP, XSL-T, JAXB, Castor XML, JiBX and JDom.
- Systems integration using IBM MQSeries, including both the IBM MQSeries API for Java and the JMS API. Familiar with programming, installing, and administering a variety of JMS providers, including IBM MQSeries, Oracle AQ, Active MQ, OpenJMS, JORAM, SunOne Message Queue, and SwiftMQ. Hands-on experience implementing server-session pooling.
- WebLogic 8 and 9 administration, including configuring clustering. Experience with IBM WebSphere (3–7), OC4J, Jakarta Tomcat, and Jetty.
- Experienced in designing and implementing modular build systems with inbuilt quality control mechanisms using Ant 1.5–1.7.

- Good experience with quality control and testing tools, including JUnit, YourKit, JProbe, hprof, OptimizeIt, CodePro Code Audit, Clover Code Coverage, and Sun DocCheck Doclet.
- Excellent experience with SVN, Jazz, Rational Team Concert, and CVS version control systems, including Eclipse integration. Some experience with Rational ClearCase and with CMVC, including authoring a CMVC plug-in to provide Eclipse 1.0 and WSAD integration.
- A background in helping others improve their technical skills: Mentored junior and mid-level programmers; published a variety of articles including for IBM DeveloperWorks; served as a technical interviewer for American Management Systems and as an instructor on Java fundamentals for American Express; participated as a presenter in several J2EE Architecture forums at American Express.
- Excellent communication skills and a strong desire to develop high-quality software.

Employment History

Google   August 2010 – present
Software Engineer Business Systems

Working to organize the world's information and make it universally accessible and useful.

IBM Research, Almaden   July 2009 – August 2010
Independent Consultant

Served as joint-architect for the design and development of the Solution Definition Manager (Global Pricing and Estimating), an Eclipse 3.5 RCP-based cost and price estimation tool for IBM Global Business Services Division.

- Responsible for analysis and implementation of business requirements, as well as fixing defects. Implemented twelve new business requirements and fixed over one hundred bugs in the application.
- Wrote a customized viewer implementation for the Nebula Grid widget, similar in functionality to the standard JFace tree viewer.
- Wrote an enhanced event-dispatching framework for the application model, with features including: robust error logging and recovery, event batching and coalescing, multiple-event delivery, and automatic listener profiling.
- Introduced JUnit-based unit testing strategy and assertion-based design-by-contract programming to the team work process. Added custom annotations to document client and server-side model relationships and wrote automated metrics using them.
- Extensive use of Rational Team Concert and Jazz source control.
- Use of the Nebula Project's supplemental custom widgets for SWT.
- Extensive use of SWT and JFace for Eclipse 3.5.
- Mentored junior team members and hosted knowledge sharing and client demonstration sessions.

SSS Blue Cross Blue Shield   March 2010
Independent Consultant

Programmed JAX-RPC web services to support Blue^2 4.1.1 multi-plan integration. Programming was performed using Rational Application Developer 7.5 and the services were deployed to WebSphere Application Server 6.1 running on Java 6 under Windows Server 2003. Provided full documentation for the solution using javadoc, and performed knowledge transfer to SSS.

SyncLive   October 2009 – February 2010

Independent Consultant

Fixed critical video streaming issues in production environment, including issues related to Flash Media Server configuration and CDN integration from a custom flash-based live video player. Assisted in capacity planning and analysis. Designed, configured, and deployed new UAT and production environments.

Dot Growth, LLC and Acadia Optronics, LLC   September 2009 – October 2009

Independent Consultant

Worked on design and implementation of a proof-of-concept SharePoint-based scientific collaboration site for the US Department of Energy. Responsible for server design, for security integration with SunOne LDAP running on CentOS, and for overseeing custom workflow implementation in C#.

Florida Department of Children and Families   August 2009 – November 2009

Independent Consultant, Chief Architect

Chief architect for an award-winning (CES 2010), first-of-its-kind solution to better track children in state custody. As chief architect I was responsible for all decisions pertaining to programming the solution, for design and documentation, and overseeing and participating in implementation with a team of four other developers.

- Designed BlackBerry and laptop mobile solutions.
- Features full GPS integration (JSR 179 and RXTX), camera integration (BlackBerry application APIs and JMF), offline support using a Google-Gears based design (using Apache Derby and for the laptop client only), form printing capabilities (StringTemplate 3.2 and Apache FOP), easy deployment and auto-updating capability using a custom-built solution for the BlackBerry and Java WebStart for the laptop client, and server-side integration using web services.
- A focus on security throughout the application due to the sensitive nature of the data:
  1. Use of the industry-standard Java Cryptography Architecture for all cryptography.
  2. Client-side data is stored separately for each user of the tool. All data is encrypted with a secure, 256-bit key that is different for each user's data.
  3. Password hashing to enable offline login uses Government-spec PBKDF2 with HMAC and SHA1, and at least 100,000 hashing rounds. Passwords are salted before hashing and the salted hashes are stored within the encrypted user data store.
  4. Passwords of any sort are not retained in memory beyond their required use period and are overwritten to secure against exploits based on analyzing memory dumps.
- Exceptional support for a wide range of BlackBerry platforms including models 7100, 7130, 7520, 8100, 8110, 8120, 8310, 8330, 8350, 8700, 8703, 8830, 8900, 9000, and 9630.
- Additional use of Eclipse 3.5, BlackBerry JDE, fledge, Java 6, Bouncy Castle cryptography, code signing certificates, Swing. Coded custom BlackBerry UI widgets for optimized data display.

ICS-NETT   May 2009 – June 2009

Senior Application Developer

Involved in design and development of the US Census Bureau's Paper Based Operations (PBO) system for the 2010 Decennial Census. Technical accomplishments include:

- Integration of the JBoss Process Manager (jBpm) into a message-driven architecture to track delayed shipments.
- Provide proof-of-concepts for several technologies including: EJB Timers, OC4J "cron" timers, and message driven beans.
- Configure and deploy Oracle AQ queues and topics.
- Designed and implemented an interactive console for benchmarking throughput of JMS providers. Used the console to obtain throughput figures for Oracle AQ and ActiveMQ.
- Assisted in creation of Ant scripts to automate the build routine.

California Institute of Technology, Heath Group   February 2009 – April 2009
Independent Consultant

Responsible for requirement analysis, design, and implementation of a tool to automate correction of acquired imagery from clinical trials of a new early cancer detection technology. The average time taken for a researcher to correct a single image dropped from two hours to ten minutes. Also the accuracy of the correction was improved which, in turn, improved the overall data quality.

From a technical perspective, implementation entailed extensive use of Java 6 and Swing, including implementation of a customized image Viewport, affine transformations, and custom Canvas-based graphics using Java2D and the Java Image I/O library.

CGI, Inc.   October 2008 – May 2009
Independent Consultant

Designed and implemented a prototype AJAX-based forms processing system using the Google Web Toolkit.

- Google Gears was used to implement offline form entry and for data caching, with extensive use of the LocalServer, Database, and Desktop components of Gears.

- Designed and integrated Java applets to support online camera capture with automatic GPS position tagging. The applet pushes captured images into the LocalServer database to make it accessible to the application.

- Designed several custom GWT widgets including a frame widget, a multistate button, JFace-style title page, a custom tab-panel implementation, and Internet Explorer 8-style accelerators.

- Implemented a Windows Vista look-and-feel for the system's dialog boxes, popup-menus, and autosuggest facilities.

- Extensive use of deferred binding to produce a smooth experience across Firefox, Internet Explorer, and Google Chrome browsers.

- Image resource bundles were used to minimize bandwidth costs.

CGI, Inc.   October 2006 – August 2007
Independent Consultant

Sr. Java EE consultant for Florida Safe Families Network (FSFN)—a large welfare automation project servicing approximately 70,000 transactions per hour. Responsible for system integration, security integration, team mentoring, and infrastructure development.

- Responsible for integrating FSFN authentication with IBM Tivoli Directory Server. Designed and implemented a JNDI-based solution using LDAP custom controls to parse extended server response codes. The solution also implemented a custom pooling solution that was stress tested to over 5,000 authentications per minute.

- Designed and implemented a single sign on mechanism for FSFN and the Business Objects-based reporting subsystem. Designed and supervised implementation of data synchronization logic between the two systems.

- Involved in design and implementation of auditing infrastructure, whereby the details of any transaction in the system can be recalled for later analysis.

- Extensive infrastructure development, including:

  1. Configuration of WebLogic 9 server environments, include clustered environments. Extensive work with node manager and Weblogic plugin for Apache. Implemented automatic pool monitor/re-starter tool using JMX.

  2. Developed and implemented improved standards for build versioning. Automated a variety of build, deployment, and code delivery tasks, including JSP compilation using JSPC for WebLogic. This involved extensive use of Ant 1.7 and bash scripting.

  3. Extensive configuration of Apache, including configuring proxies, reverse proxies, installing PHP, and the Apache Tomcat Connector.

  4. Created and administered MoinMoin 1.5.7 wiki to track environment information and to serve as a general developer wiki.

  5. Designed and implemented backup jobs and automated system checks using cron.

- Collaborated on design and implementation of search modules, including person search.

  1. Double metaphone (SOUNDEX), nickname, wildcard, and exact searches based on first, middle and last names. Over twenty available search criteria including a variety of demographics such as age and address.

  2. Round trip time of under 10 seconds for complex searches returning 100,000+ results, and near-instantaneous results for most common searches. Achieving this level of performance required extensive manipulation of clustering and non-clustering indexes on the DB2 8.1 mainframe system, as well as work with stored procedures, JDBC directional result sets, and absolute cursor positioning.

- Designed and implemented various data migration tools in Java.

- Designed various business transactions using JSP, Struts, Servlets, and Hibernate.

- Tasked with ensuring that all developers were able to work productively on their topics. This mainly involved helping developers troubleshoot Struts, JSP, and JavaScript issues, and improving their knowledge of these languages and frameworks.

Freescale Semiconductor   August 2006 – September 2006
Independent Consultant

Collaborated on design and development of Eclipse 3.2.x-based port of Metrowerks CodeWarrior's command window functionality.

- Implemented a wide variety of shell features including tab-based auto-completion, auto-completion of file system paths, scrollable command history, user-configurable font face and color, and custom key bindings. This required, among other things:

    1. Implementation of custom SWT layouts.

    2. Extensive use of JFlex for lexical analysis of output data streams for auto-coloration.

    3. Changes to standard image conversion process to preserve alpha values when moving between Sun J2SE and SWT image formats.

Webalo   January 2006 – August 2006

Sr. Software Engineer

Responsible for research and development of Webalo's suite of Java-based, web-service based mobile middleware.

- Implemented a Java MIDP 2.0, CLDC 1.1-based "Webalo User Agent". The Webalo User Agent resides on a user's mobile device and allows interaction with enterprise data exposed through web services.

    1. Conducted research of the U.S. mobile phone market to (a) determine compatibility of the MIDP Webalo User Agent with phones currently in use, and to (b) determine if simple changes to the technology stack could improve its compatibility.
    2. Implemented a custom data-grid component, optimized for low-resolution devices, with advanced features such as automatic layout, tooltip support, column selection, and drill down support. Conducted extensive testing against a Samsung A880 device.
    3. Implemented a high performance, memory efficient XML parser suitable for use in an embedded environment, using JFlex 1.4.1. The parser has a smaller codebase, smaller memory footprint, faster parsing times, and fewer bugs than the older, hand-written recursive-decent parser.
    4. Focused on optimizing distribution footprint of the Webalo User Agent without sacrificing compatibility. Ultimately reduced size from 1.2MB to 270KB using an extensive tool chain including Pngcrush, ImageMagick 6.2.9, ProGuard 3.0, and 7-zip.
    5. Extensive use of Sun's Wireless Java Toolkit 2.3 and 2.5, EclipseME, and Antenna House J2ME extensions for Ant. To a lesser extent, performed testing of the Webalo User Agent using MIDP toolkits from Samsung, Motorola, Sprint, and Palm (Treo 650 and 700 series).

- Designed and implemented JMX-based system monitoring for the Webalo Mobile Dashboard. Also, implemented a web-based front-end for accessing system monitoring information.

- Worked on implementation of account administration functionality within the Mobile Dashboard. Extensive use of Hibernate and Spring.

- Worked on the design and implementation of a Flash-based, mobile mashup service designer.

    1. Extended Webalo's existing, proprietary tool for converting Java code to ActionScript by adding support for converting inner classes, anonymous inner classes, overloaded methods, and shadowed fields. Use was made of JavaCC and The Java Language Specification, Second Edition.

2. Designed and implemented a framework for remote service invocation from the Flash client to the Java EE-based server. Significant features of the framework include an XML-based serialization format and a remote proxy generator written in Java.
3. Wrote Flash-based UI Framework on which the various mashup-related wizards that comprise the product are implemented.
4. Implemented a library of XML functions with an API that is compatible in both Java and Flash environments.

eCredit   August 2005 – August 2006
Independent Consultant

Responsible for the end-to-end design and implementation of a visual authoring environment for commercial credit and collections processes.

- The authoring environment is implemented atop the Eclipse Rich Client Platform (Eclipse RCP), version 3.1, running Java 5.0. Design and development are bridged through the use of a model driven architecture, using EMF, version 2.2. The visual editor component of the system is implemented using the Graphical Editor Framework (GEF).
- Business processes are serialized into XML format for easy interchange with other systems.
- Implemented a Windows-based installer using the Nullsoft Scriptable Install System (NSIS).

Ohio Department of Taxation   June 2005 – August 2005
Independent Consultant

Consulted on the design and implementation of a J2EE-based taxation system for handling taxpayer registration, returns, and workflow requirements. The system has the following characteristics:
- DB2 v8.0 was used as a data store and triggers were utilized extensively to implement data auditing requirements.
- Business logic was implemented within WebSphere 5.0 using EJB session and entity beans.
- A standard Model 2 architecture was implemented utilizing Jakarta Struts and the Tiles Document Assembly Framework.

Expertise was also provided in the following areas:

- Implementation of a data access service. Wherever possible, the layered service implements component design principles such as inversion-of-control. Key features include:
  1. Automatic data pagination, which is a key to maintaining service level requirements.
  2. A scheme to decouple the work of query authors from the user interface designers.
  3. All aspects of a data access instance (query) are located in a cohesive, class-based unit.
- Reviewed logical schema model. Provided feedback for indices, normalization opportunities, as well as general enhancements in the light of business requirements.
- Design of database population and deletion scripts.
- Definition of a strategy for automated quality assurance using Load Runner.
- Extensions to Struts custom tags to improve productivity of the user interface team.
- Training and mentoring developers.

NYSE, New York Stock Exchange   February 2005 – May 2005
Independent Consultant

Served as a senior consultant for the design and implementation of a fraud tracking system. Business logic was implemented within WebSphere 5.0, while the front-end rich client was implemented atop the Eclipse platform. Expertise was provided in the following areas:

- Formulated a JUnit-based unit testing strategy for server side business components, and provided an initial proof-of-concept.
- Proposed procedures for basic incident tracking, as well as procedures for utilizing JUnit for regression testing of incidents. Provided a proof-of-concept system using Mantis 1.0.0 running on Apache HTTP Server and MySQL.
- Audited logging procedures within server-side business logic and made several proposals, including the use of Log4J nested diagnostic contexts to improve the ability to correlate logging statements.
- Audited server-side security model and issued several recommendations.
- Implemented a rigorous type system to increase front-end developer productivity and provided a detailed, three-stage roadmap for its evolution.
- Performed a comprehensive audit of the rich-client tier of the application and issued a findings report. Also performed a small-scale feasibility analysis on the use of the Eclipse Modeling Framework to improve productivity.
- Designed and implemented a data grid component to standardize display and manipulation of tabular data within the system. Key features include:
  1. A user interface modeled after Microsoft Excel and Microsoft Access. Features of the user interface include column reordering, column show/hide, record-set navigator, selection indicators, and row numbering.
  2. Support for multi-column sorting, multi-column filter specification using Boolean sheets and supporting a variety of match operators including regular expressions.
  3. Support for persistent sort and filter profiles, column ordering, and column widths using the `java.util.prefs` API.
  4. Microsoft Excel data export.
  5. A developer API that greatly simplifies loading data stored in Java Beans.
  6. Data formatting is tightly integrated with the application's type system.

eCredit   September 2004 – January 2005

Independent Consultant

Served as a senior resource in the design and implementation of the Mobius Process Development Environment (PDE). The PDE, which serves as a business rules and process flow authoring system for the Mobius business process engine, is provided as a feature for Eclipse 3.1 and Java 5.0.

- Designed and implemented a shared workspace using Jakarta Slide 2.1 WebDAV repository. The client view of the repository is represented using an EMF 2.1 native ECore model that is loaded and persisted during plug-in activation and deactivation, and provides support for workspace synchronization, asynchronous deep and shallow refresh, locking and unlocking of resources, check-in and check-out capabilities, and a drag and drop operation for resources. In addition, a WebDAV-compliant recycle bin was implemented. Etag-based content caching was implemented to improve workspace performance. Responsiveness was increased through heavy use is made of the Eclipse 3.0 Job API.
- Designed and implemented a secure licensing model for the PDE. Licenses, which contain roles and their expiration dates for a specific principal, are obtained from a licensing server after providing a valid license key. Licenses are signed by the licensing server to prevent modification and encrypted using the PDE client's public key to prevent unauthorized access.

Implementation entailed the use of strong (2048 bit) RSA-based X.509 certificates for license signing and symmetric key encryption, as well as the AES symmetric cipher (Rijndael variant) for bulk data encryption. Heavy use is made of the Sun JCE implementation and the Bouncy Castle 1.25 JCE implementation.

- Designed and implemented a runtime view of business process servers. Wizards were included for adding new servers, and, existing servers can be visually explored and operated on. For example, servers can be started and stopped, new business processes can be deployed on particular servers, and runtime logs for particular processes can be opened. The client view of the runtime environment is represented as an EMF 2.1 annotated Java ECore model. The view provides Job-based asynchronous refresh for enhanced responsiveness.
- Implemented web services-based integration with business process servers. This involved the use of SOAP, SAAJ, and JAX-RPC, including the use of wscompile for client-side stub generation.
- Implemented editors and viewers for several types of XML documents in the Mobius system. Models, edit frameworks, and basic editors were automatically generated from W3C schemas using EMF 2.1. Editors were then heavily customized to include
  a. Master-detail block support.
  b. Writeable and read-only modes that are visually indicated through a customization to the toolbar. Read-only mode is automatically enabled whenever a file is opened but not locked by the current user,
  c. Automatic upload to WebDAV repository as part of the save sequence.
  d. Enforcement of semantic constraints using problem markers for edited files.
- Designed and implemented five wizards using the JFace Wizard framework.
- Provided Eclipse mentoring for eCredit employees assigned to the project.

Islamic Society of Greater Columbus   February 2004 – June 2004
Independent Consultant

Provided pro bono consulting expertise towards the development of a web-enabled membership and donation database. This system served to streamline the workflow of the organization and represented an upgrade from the old fat-client system written in Microsoft Access, which suffered for many problems including data synchronization and an inefficient user interface. In addition, a variety of reporting and mass mailing features were added.

- The data tier was implemented using the Firebird 1.5 RDBMS. The schema was normalized to the fifth normal form (5NF), and synthetic keys and indices were added to optimize performance. Name searching was optimized through indexed soundex columns in the individual entity information table.
- A data converter and cleanser was implemented to move data from the old data tier (Microsoft Access) to the new one (Firebird).
- The application tier utilized the model-view-controller architecture pattern and was implemented using Java Servlets and Java Server Pages. Data access utilized the *data access object* pattern.
- The presentation tier made heavy use of CSS2 to minimize data transfer requirements while maximizing cross browser portability of the application.
- The application was deployed as a WAR file to Apache Tomcat 5.0, running on a Windows XP system and was tested using both Internet Explorer 6 and Mozilla Firebird 0.8 browsers.

CGI-AMS   August 2003 – June 2004
Independent Consultant

Consulted as an architect for the Office Field Audit Support Tool (OFAST) project, a large project for the Ohio Department Taxation employing approximately twenty-five full-time staff, involving the design, development, testing, and deployment of a Swing-based auditing tool used by tax auditors across the state. Tax rules were coded in a custom language optimized for rules processing.

- Responsible for designing and implementing the architecture of the OFAST user interface. The architecture has the following key components:
  1. Screen definitions are stored in XML files and are dynamically built at runtime from those definitions. A W3C XML Schema, verified using Sun's MSV 1.5 and IBM Web-Sphere Studio 5.0 schema validation facilities, was authored for the screen definition language.
  2. A data grid type whose values can be bound dynamically to an entity array. The data grid also supports an Excel 2000 look and feel, dynamic column sorting with visual indicators for ascending, descending, and default sort orders, line numbers, multiple layers of transparency, cell editors and renderers that vary from row to row in the same column, and data cells whose values can be dynamically bound to reference data groups stored in the database (and are hence selectable through a combo box).
  3. An extensible strong-typing system for managing user input. A type contains operations for data validation, data parsing, and data formatting, as well as operations for returning a default renderer and editor. A type's operations are strictly defined mathematically thereby enhancing flexibility and maintainability.
  4. A form header component that allows forms to be associated with a suggestive image, a title, and instructions. Instructions can be dynamically modified at runtime with warning and error messages.
  5. A wizard framework to allow complex, sequential operations that comprise many logical pages to be developed rapidly. Built in type validations.
- Architected and implemented the user interface and business logic tiers for the Corporate Franchise Tax audit type. Additionally, participated in functional requirements analysis as well as functional design activities.

  Tax forms, which constitute the basis for an audit, vary from year to year and are dynamically generated from database metadata information. In addition, corporate tax forms are inter-linked, receiving values from one another during the course of computations. The user inter-face makes heavy use of Swing, especially tables, and provides a rich set of features to sup-plement standard data entry operations, including, but not limited to: visual indicators for overridden, sourced, and calculated fields; line and form-level notes, cell-specific tooltips to indicate sourcing relationships; dynamic, multi-layered line shading to indicate selected lines and lines with discrepancies; custom navigation features including navigate-to-source functionality.
- Mentored developers and introduced more rigorous quality control procedures into the application development lifecycle, including: implementation of an Ant 1.5 and 1.6-based modular build system, W3C Schemas for all XML document types, and a number of standalone quality-control programs written using Jakarta CLI.
- Designed a visual editor for creating and editing user interface definition files using the Eclipse Modeling Framework 2.0 and W3C XML Schema 1.1.
- Designed and implemented an integrated development environment for the custom rules proc-essing language in Java Swing. The development environment provides the following features to developers:

1. File and file set loading for entity definition files, trace files, user interface definition files, and decision table files as well as the ability to export and import file sets.
2. File viewers for each of the file types, including syntax highlighting (using the JEdit Syntax Package) and tree views for XML files. The trace file view includes integrated step-into support to restore the state of the rules engine to a given point in the program execution.
3. An entity view to support viewing individual entities and their metadata.
4. A three-page wizard to guide users through the process of creating new decision tables. This wizard includes sophisticated regular expression checks (using Jakarta ORO and Perl 5 regular expressions) on the new decision table, as well as a number of other integrity checks.
5. A table editor component consisting of functions for compiling, deleting, saving, and validating decision tables, as well as four sub-screens for providing access to table data. The component also automatically logs and tracks modification histories for every table, in addition to performing transparent validation of table structure. The first screen provides access to table metadata as well as read-only access to the audit log for the table. The second screen supports table modifications and includes an editor that supports auto format plus diagnostic compile features. The third screen supports a call-to view, showing which tables are called from the current decision table. Finally, the fourth screen provides read-only access to the raw XML source for the table. Manipulation of the decision table document was accomplished using DOM 3 features in Xerces.
6. A feature to export documentation for all decision tables, including comments and formal declarations, into hyperlinked HTML files for easy viewing in a browser.
7. An expression evaluator for dynamically executing actions and conditions within the current context and displaying the results.
8. Stack explorers that allow the entities on the rules engine's runtime stacks to be recursively explored, starting from either the data or entity stacks. Every entity's attributes, including array, primitive, and entity-type attributes, can be inspected.
9. A navigator view that allows convenient access to all resources loaded into the IDE. The navigator view dynamically updates as resources are added or removed, and provides support for decorators to further indicate the state of the loaded resources. For example, active resources use a bold font, unsaved resources include a floppy disk decorator and an asterisk after the name, and invalid decision tables have a yellow warning sign decorator.
10. User experience optimized for JDK 1.4, but supports graceful degradation under JDK 1.3 using reflection.
11. Support for internationalization through the use of resource bundles.

- Designed and supervised implementation of POI-based Microsoft Excel integration to replace existing Jawin COM-based integration. Resulting implementation was an order of magnitude faster than COM-based implementation and used considerably less memory as well.
- Provided two four-hour training seminars as well as a number of shorter training lectures for state employees. The purpose of the lectures was to provide junior-level programmers with the necessary information to begin programming in the OFAST environment.

American Express   May 2002 – August 2003

Independent Consultant

Worked as a senior developer and junior architect within the Architecture Team of American Express's Interactive Technologies Division.

Standards Governance

- Involved in authoring several internal, strategic, position papers regarding J2EE strategy within American Express. For example, I was involved in defining a strategy for enterprise services.
- Designed and taught key parts of an internal Java training class for American Express Employees. Developed a curriculum consisting of lectures, programming assignments, and group exams to prepare students for the Java Programmer Exam.
- Lectured on Java, XML manipulation technologies, and data binding at several Architecture Forums and Java Forums.

Consulting

Provided consulting to application teams. Generally consulting requests fell into one of three categories. Overall, dozens of consulting requests were addressed.

- Troubleshooting critical production problems in J2EE applications.
- Determining performance bottlenecks within J2EE and internet applications and providing guidance in resolving these issues to application teams.
- Assisting application teams in the implementation of American Express and Industry J2EE standards through the use of code reviews and mentoring.

Component Development

- Architected and oversaw the implementation of several new infrastructure components. These components are being used successfully by applications with some of the heaviest loads within American Express, including one application that handles over ten million transactions daily, and several that handle at least one hundred thousand transactions daily.
- Designed and implemented a major functionality upgrade to American Express's standard web application controller. This entailed the following tasks:
  1. Provided UML sequence and class diagrams to illustrate the new system design.
  2. Rewrote the old code base(s), structuring the new code base using industry standard design patterns.
  3. Supplemented the code base with a test suite authored using mock objects and JUnit.
  4. Authored a DTD for the new configuration file format and wrote a configuration file conversion utility that utilizes JAXB to simplify coding.
  5. Rewrote user manual and provided a supplementary design manual authored in LaTeX.
  6. Provided Solaris migration scripts written using Korn Shell and AWK.
- Served as architect and technical lead for a service activator projects. Service activator provides a high-quality message-driven bean container for pre-EJB 2.0 application servers.
  1. Provided UML sequence and class diagrams to illustrate system design and operation. Presented ideas and obtained buy-in from senior architects.
  2. Implemented server session pooling to improve container scalability.
  3. Oversaw testing with a wide variety of JMS vendors to ensure maximum interoperability.
  4. Provided JMX-based administration over HTTP using MX4J.
  5. Authored W3C schema for configuration file (modeled after EJB 2.1 specification for message-driven beans) and utilized Castor XML to simplify loading process.
  6. Provided advanced analysis tools for activator to provide a statistical analysis of performance that was used to assist clients in meeting service level requirements. Also analyzed performance figures for a variety of JMS vendors to determine minimum transaction lengths necessary for system feasibility (defined as at least 70% efficiency).
  7. Implemented a strict lifecycle modeled after the Avalon Phoenix Server to maximize container flexibility.
  8. Oversaw documentation and testing.

- Served as architect and technical lead for service aggregator, which facilitates asynchronous execution of processes (possibly heterogeneous back-end systems) and aggregation of results.
  1. Provided UML sequence and class diagrams to illustrate system design and operation.
  2. Designed client API, modeling after the `java.lang.Process` API.
  3. Designed and oversaw implementation of stub compiler.
  4. Oversaw documentation and testing.
- Designed and implemented a plug-in to integrate WSAD with CMVC. Provided a wide variety of features including automatic defect retrieval when performing check-in (using CMVC Report). Plug-in is currently endorsed as a standard by at least one large (120 developers) application team within American Express.

American Management Systems   August 2000 – April 2002

Developer

All work was done within the Integrated Client Management System project, a large project (200+ consultants) for the Ohio Department of Job and Family Services involving the design, development, testing, and deployment of an intranet-based case management tool for use by caseworkers throughout Ohio.

General responsibilities (ongoing)
- Serving as a Java technical interviewer to help in identifying experienced candidates for the project and for the company.
- Providing extensive Java related assistance to junior programmers.
- Assist other projects with J2EE-related architectural decisions.

CriseCom Architecture Team (3 months) – Devoted to the design, implementation, testing, and deployment of CriseCom, an Java infrastructure and API to provide business-side developers access to CRIS-E, a legacy system, from the eICMS web application. CRIS-E serves as the system of record for much of the state's welfare system, and the CriseCom infrastructure provides synchronous, parallel access to the CRIS-E system for read-only transactions, and asynchronous, sequenced, and guaranteed-delivery access for all other transactions within the system. The end result is an almost seamless integration between eICMS and CRIS-E from the perspective of a caseworker. Within this project, I assumed the following responsibilities:
- Involved in the design and implementation of the CriseCom architecture, which uses DB2 for transaction queuing and tracking, MQSeries for guaranteed delivery between intermediate processing nodes within the system, and three custom-built Java servers (processing nodes) dedicated to different phases of transaction processing.
- Proposed and helped implement an Java and XML based configuration scheme which allows the system to configure itself entirely at runtime from an XML configuration file.
- Designed and implemented the CriseCom API for Java. Programmatic access was provided to CRIS-E on a screen by screen basis, in a manner very similar to the SAP R/3 BDC API or IBM Host On Demand™ or Host Access Class Libraries.
- Designed and implemented a web based administration tool for the system using Jakarta Struts (v.1.0), a high performance Java J2EE framework.

From a technical point of view, this entailed:
- Heavy utilization of the MQSeries for Java API, as well as use of MQSeries as a whole.
- Use of Jakarta Struts (v.1.0) in implementing the web based administration tool. Action classes, action forms, validation, and error handling capabilities of Struts were used to create an administration tool capable of:

1. Authenticating users and restricting access.
2. Starting and stopping individual processing nodes on the fly.
3. Viewing and modifying the contents of queues online.
4. Ascertaining the load on the system.
5. Viewing system logs online.

- Writing a controller server to handle the execution of transactions. This involved the dynamic lookup, loading, and execution of a "Java business object" in response to a user request. To increase performance, object caching was implemented, causing a throughput increase of four-hundred (degenerate) transactions per minute.

- Writing a Java logger server based on a publish-subscribe model. Errors are extracted from completed transactions, are mapped to a set of actions (dynamically configured through a database), and events are generated for these actions. These events are then dispatched to subscribers, which are multithreaded to increase performance.

- Writing a Java-based `MQQueueManager` connection factory to speed access to MQSeries, and, ultimately, to CRIS-E.

- Utilizing IBM's Enterprise Access Builder to automate the creation of Java record objects from COBOL screen definitions.

- The creation of several utilities to assist business developers, including an automatic screen copier, an automatic data converter, and an automatic pre-image data validation utility. Development involved heavy use of the `java.lang.reflect` and `java.beans` libraries.

- Ensuring the complete and thorough documentation of all code using Java 2 Javadoc standards. Furthermore, assumed responsibility for performing a complete code-review.

- Executing stress tests and writing performance monitoring tools on the system. A throughput of approximately 7000 small transactions per hour was realized.

- Providing extensive support to business programmers, as well as preparing a manual describing the CriseCom API, examples of how to use it, and an overview of the architecture.

- Mentoring a junior programmer in writing one of the three Java servers required by the architecture.

Technical Architecture Team (14 months) – Devoted to the development and improvement of the Java framework that underlies the eICMS project. Within this team, I assumed the following responsibilities.

- Suggested, designed, implemented, and deployed a user-exit architecture within the framework that allows business logic calls to be made after arbitrary events in the servlet service cycle, besides the main business exit. The architecture provides the following features.
  1. An XML configuration file, with DTD, is provided to facilitate configuration. Options exist to enable and disable exits, to cache and multithread them for improved performance, and to configure exit execution sequences.
  2. Dynamic, runtime loading of user exits. Data is dynamically bound to the exit using the `java.lang.reflect` and `java.beans` libraries.
  3. A complete user manual which includes examples of and instructions for writing user exits, a description of adding new user exit calls to the framework, full Javadoc, and a fully commented DTD for the XML configuration file.

- Designed and implemented a pluggable security architecture. Complete documentation and an LDAP security provider are included. In addition, an XML-based provider (with a DTD and Javadoc) is available for projects with lighter security needs. This architecture has been successfully utilized by other projects to interface with existing security systems at client sites (e.g. Novell Directory Service, DAP, LDAP, RACF).

- Researched and provided a solution for accessing host systems directly through the eICMS web application. Using IBM's Host Access Class Libraries (v.5), a solution was developed with the following characteristics:
  1. Emulates a 3270 terminal session over the web using Java Applet technology.
  2. Allows the eICMS system to step a user through a sequence of screens on the host system, interacting with him/her in the process. This functionality is used to take users to a point of failure in an automatic update, and to recommend a suggested fix.
  3. Allows developers to program macros for the system using XML, and to execute them on the client using LiveConnect™ technology.
- Designed and implemented a table data structure for project wide use. Similar to the SAP ABAP table, functionality is provided to define primary keys, sort by primary key, perform an arbitrary SQL sort, perform SQL deletes, delete adjacent duplicates, find by primary key, insert rows, delete rows, request an iterator, and collect (sum) data by primary key. SQL parsing is accomplished through use of ZQL, a Java SQL parser.
- The data model for the project entails the storage of reference data as numeric codes. Previously, these codes were sent, unmapped, to the client, where their values were retrieved through calls to the server. I implemented two improvements to this model.
  1. Cached reference data within a data island on the client to improve speed and stability.
  2. Implemented a mechanism to automatically map values on the server side. Mapping is done with the aid of XML map files which stipulate target fields that must be converted. This involves heavy use of DOM 2.
- Responsible for implementing encryption facilities in the framework to secure sensitive information stored in configuration files. This involved the utilization of the Cryptix™ (v.3.2.0) Java security provider and AES-standard Reijndhal encryption.
- Responsible for decreasing client response times and network load. The eICMS system downloads an entire logical unit of work to the client machine as an XML data island. While having benefits, response times suffer on slow networks. To improve performance, a Java applet that compresses outbound data islands was designed. Base-64 encoding eliminates HTTP transport issues, and LiveConnect™ technology allows the applet to be called through DHTML. On the whole, application response time was improved by 30% over a slow (56 kbps) network.
- Designed and implemented a connection factory class that allows JDBC calls to be made easily and efficiently. Robustness and performance were maximized by taking advantage of JDBC 2.0 and connecting to a WebSphere managed data source pool using JNDI.
- Suggested the adoption of Log4J™ (v.1.1b6) as the standard logger for our project. This involved implementation within the framework as a proof of concept, suggesting an overall deployment strategy throughout the project, briefing management, and assisting business developers in its use.
- Improved the accessibility of the LDAP provider, and thereby reduced application downtime, by sending read-only requests through an LDAP dispatcher while sending update requests to a read-write copy of LDAP. Previously, all requests were sent to a read-write copy.
- Greatly improved the performance of a convenience XML class utilized throughout the project. In addition to standard optimizations such as minimizing object creation, caching, code hoisting, and delayed evaluation, the DOM was upgraded to version 2, and a variety of DOM-specific optimizations were made. JFlex™ (v.1.3.4) was used to generate a fast lexical analyzer for parsing. This resulted in 70% improvements in time and 200% improvements in memory among the Enterprise Java Beans deployed in the system (performance was measured using JProbe™).

- Designed and implemented a utility to strip logging statements directly from compiled Java code. The utility made heavy use of the Byte Code Engineering Library (BCEL).
- Enhanced performance and architecture of the security system by improving caching and implementing a table lookup mechanism.
- Responsible for preparing release notes for new releases of the framework, and providing assistance to other projects using the framework. Also, for acquainting new developers with the operation of the framework.
- Fixed many bugs within the framework, added Javadoc to several packages, and performed general performance tuning on the system.

Server-side and Client-side Business Development (4 months) – Development of Java Server Pages and Enterprise Java Beans to meet client requirements. In addition, solely responsible for technical design and coding of print functionality for a business domain within the application.

- All documents were served in PDF. This entailed writing an XSL FO style sheet incorporating XSLT to provide dynamic content, and applying it to an XML containing report data using Apache FOP (v. 0.16) on the server side.
- Printed documents were compressed, base-64 encoded, and archived in DB2 for reprinting.
- Heavy use was made of data binding, DHTML, and JavaScript on the UI.
- One Enterprise Java Bean (stateless session bean) was developed to encapsulate business functionality on the back end.

Harbor Capital Investors   March 2000 – June 2000
Independent Consultant

- Designed and implemented online HelpDesk intranet application for managing internal company bug and error reports (tickets). Java Server Pages, servlets, Java Beans, and custom tag libraries were developed and used throughout the project. Heavy use was also made of JDBC. Testing and development took place on the Jakarta-Tomcat web server.
  1. System consists of an application module with sub-modules allowing for the creation, tracking, and modification of tickets. Reporting facilities are also included, as well as an online administration module.
  2. Restricted access with different feature sets provided to administrators, help desk technicians, and end users.
  3. Provided support for English, Spanish (partial), and German (partial). Furthermore, developed custom tag library and a simple Java Swing-based tool to make adding additional language support painless.
  4. Content creation and development tasks were separated for maximum maintainability.
- Implemented class to handle a wide variety of internal rates of return in Java, including XIRR, MIRR, IRR, and PNV. Heavily overloaded all financial functions to provide maximum functionality. Class provides great control over parameters and calculates rates more rapidly than the industry-standard Microsoft Excel spreadsheet program.
- Created SQL generator class in Java to generate SQL cross tables dynamically and with great accuracy.

References

Available upon request.