



Delivering Business Logic for the Federal Government
Enterprise Architecture with Versata

> A Basis for Meeting Discussions



Introduction

The Federal IT Architecture Challenge

In Government, as in commercial enterprise, the drive for more efficient operations and better public service has renewed interest in simplifying and consolidating functions across organization boundaries and existing IT systems.

Today, information architects must design infrastructures from complex and evolving technologies to support a wide-range of new business processes. In many cases, the new process will provide an integrated, service-oriented “facade” to existing applications and data. In some cases, new business applications will be developed to support re-engineered functions.

Any development approach suggested must support a layered model, conceptual enough to be understood by business users but detailed enough to be implemented.

Within the E-Gov Enterprise Architecture Guidance documents, these layers are called out as Reference Models including the Business Reference Model, the Conceptual/Process Model, the Data Architecture Model and the Technology Architecture Model.

This sophisticated, layered architecture reflects the Industry’s most current standards and best practices. However, given the time, budget, and in some cases, skills constraints facing Government IT shops, delivering systems against this architecture may be a challenge.

Versata’s Value

Versata can help meet this challenge by creating new business functions without having to write major pieces of code. It allows business processes to be graphically designed, allows transactional details to be declared using high-level business rules, and automatically constructs and deploys components that execute in COTS application servers. It reduces development time, increases system quality, decreases maintenance costs and leverages existing development skills.

How Versata Works

The Role of the Application Server

Within the E-Gov Enterprise Architecture Guidance documents, J2EE application servers and related technologies are cited for their potential to meet many components of the Technology Architecture.

J2EE (Java 2 Enterprise Edition) application servers have become the de facto standard for hosting internet-available applications on a variety of hardware platforms. The layered services and containers of the application server support the layered Enterprise Architecture and cleanly separate business logic from user interface components or system integration concerns.

The Role of Business Logic

Within the E-Gov Enterprise Architecture Guidance, business logic can span the Business, Technical and Data Architectures.

The business logic of an enterprise is made up of all of the evaluations, decisions, transitions, transformations, requests and responses needed to carry out its organizational functions. Essentially, this logic is the “value added” by the people and systems of the business. The function of enterprise information systems is to capture this “value add” and enact it in automated processes.



Business logic itself is conceptually layered and can be generally divided, from highest to lowest layer, into:

- **Process Logic** - Logic that moves work between a series of stages or steps. For instance, when a citizen applies for Government assistance, the process may move from identification to application to approval. At each step, activities may be performed by humans or systems and steps may be transactional or non-transactional
- **Transaction Logic** - Logic that processes data to move it between consistent states at a point in time. For instance, when submitting an assistance request, a transaction may be used to process form data and create an application record.
- **Data Logic** - Logic that derives, validates or otherwise constrains data values to enforce operational policies. Data logic may also be responsible for consolidating or reconciling multiple data sources into a consistent enterprise view of enterprise data. For instance, when submitting an application, a citizen's Social Security number may be validated against one agency's data, while annual income may be validated against another.

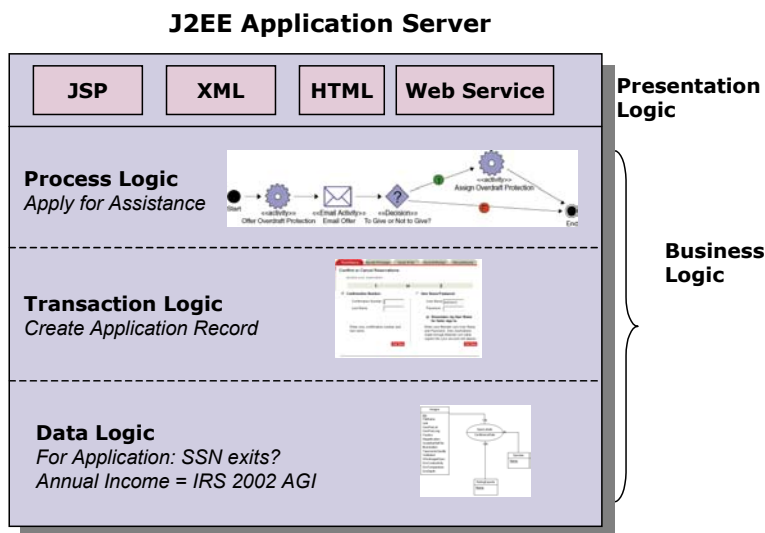


Figure 1 - Business Logic Layers

Ideally, logic in each layer should be captured as clearly and simply as possible. Elevated from the source code of applications, business logic can become an understandable, sharable, reusable and maintainable asset that spans generations of systems and development teams.

Similarly, the business logic should be implemented as automatically and efficiently as possible, so that systems are achievable in available time and with available personnel.

Versata's Role – Logic Development, Execution, Management

Versata provides the most productive way to create, execute, change and reuse business logic within J2EE application servers. Versata uses declarative programming techniques to capture the business logic of business functions as transactional rules and process models in the Transaction Logic Designer and the Process Logic Designer. At the metadata layer, the rules and models are expressed in XML.

Within the E-Gov Enterprise Architecture Guidance, XML is a common interchange format for all levels of the architecture.

Logic is compiled into extensible Java components that are executed by logic engines that reside in the EBJ container of the J2EE application server.



Transactions and processes can integrate data using a variety of technologies including: Java Transaction Services (JTS), message brokers and XML-based WebServices. They can be accessed from nearly any client-interface including HTML, JSP's, and through XML-based WebServices.

Within the E-Gov Enterprise Architecture Guidance, JTS, message brokers and XML-based WebServices are called out as Technical Models.

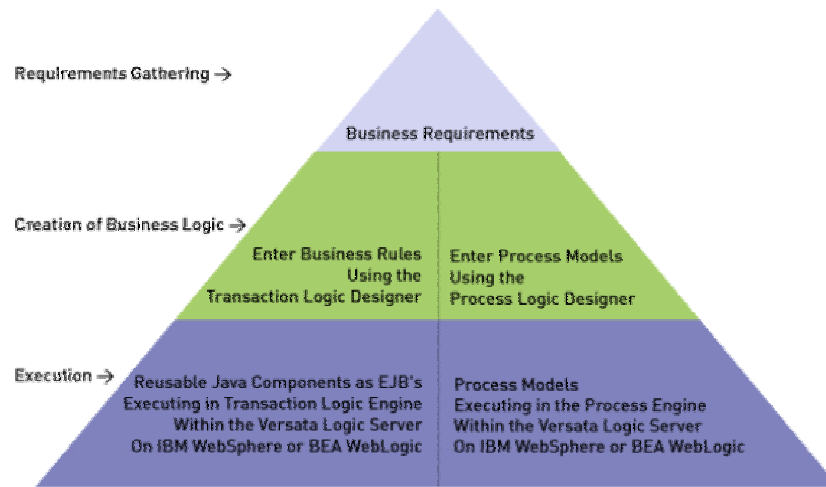


Figure 2 – Capturing and Executing Business Logic



Conclusion – Focus on Function

Automatically Deliver to Technology

For Government, declarative business logic represents the next step forward to bridge the world of business function with the world of optimized implementation.

The approach uses a small number of accepted, non-technical concepts that can be shared between operational and technical teams. Operations managers, who may not be interested in architectures, can be engaged in the design of policies, steps and business rules. Furthermore, technical teams, who may lack time or J2EE expertise, can consistently generate systems that automatically meet Architecture Guidelines for scalability, flexibility and interoperability.

Through a broad and integrated approach to all types of business logic, Versata enables IT solutions where the business function, not the technology, is the focus.



Versata, Inc. 300 Lakeside Drive, Suite 1500, Oakland, CA 94612 USA
web www.versata.com toll-free 1.800.984.7638 ph 510.238.4100 fx 510.238.4101

United Kingdom
Versata
Parkshot House
5 Kew Road
Richmond
Surrey TW9 2PR
England
ph +44 (0) 20.8334.8080
fx +44 (0) 20.8334.8180

France
Versata
13, rue Camille Desmoulins
92441 Issy-les-Moulineaux
France
ph: +33 1 58 04 27 87
fax: +33 1 58 04 26 27

Germany
Versata
Flughafenstrasse 52
22335 Hamburg
Germany
ph +49(0)40-53326-0
fx +49(0)40-53326-100

© 2003 Versata, Inc. All rights reserved. Versata, Versata Logic Server, and Versata Studio are trademarks of Versata, Inc. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other company and product names mentioned are the trademarks or registered trademarks of their respective companies.

