



Gen2SOA: Program Overview

January, 2009

2750 Prosperity Ave, Suite 210
Fairfax, VA 22031

703-246-0000 • 888-383-7927
www.Everware-CBDI.com

© 2009 Everware-CBDI, Inc.
Version 2.7

Introduction

Information Technology (IT) organizations today face a variety of challenges in achieving their missions. The fast-paced global business environment requires organizations to continuously reevaluate and update their business processes and supporting technical infrastructure to have any hope of keeping pace. Competitive pressures drive businesses to open their processes to customers and partners and to move their service interfaces closer to the source or destination, whether inside or outside of the company. But massive rebuilding of core business functionality, embedded in legacy applications, is simply too expensive, takes too long, and is too risky. In many cases, the most effective solution is to enable legacy applications to interoperate with new technologies, thereby quickly providing new capability without needlessly replacing what is working.

The pressure to quickly provide internal and external users with Internet accessible, browser-based applications is felt by most businesses in today's environment. Emerging technologies, such as Java, .Net, web services, application integration, and component-based development, require organizations to choose from a myriad of platforms, new products, and new development standards. Faced with constrained IT budgets, businesses must leverage their IT investments in people, technology, and systems to meet these challenges.

For organizations that have made large investments in CA Gen, the inventory of Gen systems supporting core business requirements is typically substantial. It is critical that this investment be fully leveraged in this "new world"... a world centered on the Internet and Service Oriented Architectures (SOA).

To address this need, Everware-CBDI has developed **Gen2SOA**, a program consisting of development processes, methods, and consulting services designed to exploit the new and emerging capabilities of CA Gen in the SOA environment. Part of Everware-CBDI's Legacy Modernization Program, **Gen2SOA** provides a rapid, multi-dimensional path from the Gen legacy transaction environment to the Internet using today's leading service-based technologies: Java and .Net. This program greatly reduces the time and effort required to create new web-based service oriented applications, while leveraging the investment in Gen enterprise applications. Everware-CBDI's **Gen2SOA** program enables organizations to quickly utilize elements of existing Gen enterprise applications in a variety of deployment scenarios.

Customer Trends Drove the Creation of **Gen2SOA**

CA Gen customers are not immune to these business and technology drivers. As Gen service providers for many years, Everware-CBDI has observed some important trends that inspired the creation of **Gen2SOA**.

- A pervasive shift to web-based application deployment has taken place with the Internet becoming the integrating platform of choice.
- There is a clear and understood need for an architected approach, but often this is deferred in the urgency of the moment to do it “quick and dirty” and clean it up “later”.
- There is a compelling requirement to continue to reuse software assets in which organizations have invested significant financial and intellectual capital.
- An emphasis on business agility and centers of business processing, both internal and external to the business.
- There is an ongoing and accelerating requirement to rapidly adopt newer or emerging technologies and architectures as a competitive business advantage.
- And, as always, this all has to be accomplished while at the same time enhancing:
 - Time to market
 - User Satisfaction
 - Effectiveness of IT dollars
 - Application maintainability
 - Staff retention and motivation.

Clearly, an integrating framework and a transition plan with options are needed: A clear roadmap to guide decisions, planning, and execution. Hence, **Gen2SOA!**

Gen2SOA – Description

Gen2SOA is a comprehensive framework for successful modernization of Gen legacy assets to the web and services based environments. In this rapidly changing environment, every organization has a unique set of requirements based on organizational size, current application portfolio, technology platforms, implementation timeframes, business requirements, budgetary constraints, etc. **Gen2SOA** consists of a flexible structure of tasks, techniques, and deliverables that accommodate a wide variety of requirements while clearly defining a roadmap to success.

Gen2SOA consists of four phases that are executed iteratively to realize the Customers vision for their modernization program:

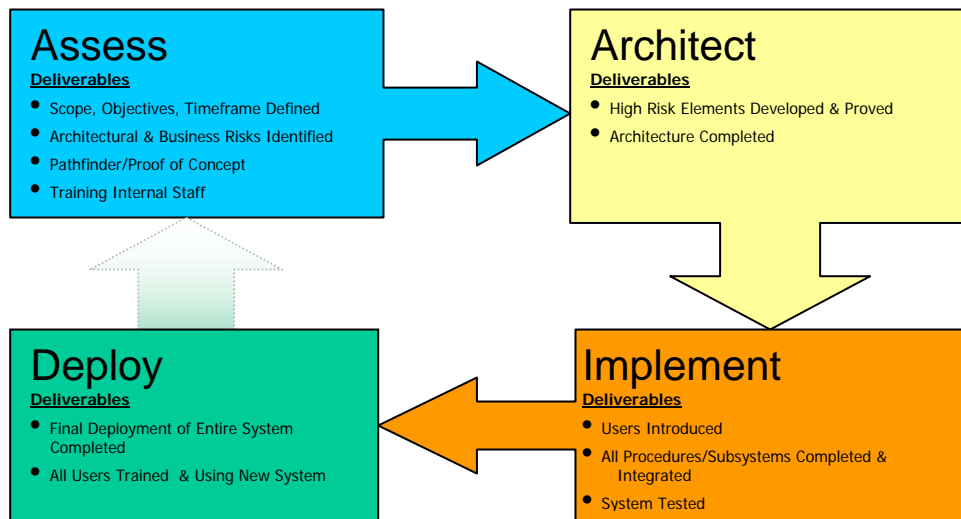
- **Assessment** – Early iterations establish the baseline for evolving the organization’s Gen applications to the web and services environment. The initial iterations often include a Pathfinder or Proof-of-Concept project. If training and knowledge transfer is required, it is largely accomplished during this phase. In each assessment iteration we review the current situation and determine the objectives, timeframes, and major risks for the iteration as we progress towards the final vision.
- **Architecture** – Builds upon the Assessment to create and elaborate on the business / technology framework, or blueprint, for the **Gen2SOA** program. High-risk areas identified in Assessment are addressed and steps are taken to mitigate risk.
- **Implementation** – Modernization activities focused on producing the elements of the final SOA, examples may be wrapping services, harvesting services, porting/re-platforming services, and developing composite application functionality.
- **Deployment** – Completes the move of all elements delivered during the current iteration into production usage. This phase includes data conversion and end-user training when appropriate.

The first step in each iteration through the phases is to establish clear objectives for the iteration in the context of the *customer’s* **Gen2SOA** vision, as determined in the initial Assessment. All subsequent planning and execution is then driven by and measured against the iteration objectives within each phase. Each iteration is designed to reduce risk for the overall

project, by producing specific deliverables that facilitate the subsequent iterations.

The **Gen2J** Lifecycle

Each phase in the **Gen2SOA** lifecycle can be thought of as a “mini-lifecycle” unto itself, but each with a different emphasis and set of deliverables.



Thus, **Gen2SOA** provides the needed systematic approach to making sound business decisions and then accomplishing business and technology objectives in optimum fashion. **Gen2SOA** facilitates the transition of legacy Gen applications to an SOA, enabling these applications to interoperate at various levels with newer, emerging technologies, e.g., Java and .Net, achieving maximum leverage of IT investments. **Gen2SOA** provides an informed evaluation of options, and an iterative approach to planning and implementing the move towards SOA.

Gen2SOA Implementation Strategies

Gen2OA focuses on three key strategies for deploying existing Gen applications within an SOA. These strategies are not mutually exclusive, in fact, a combination of strategies would typically be implemented as an organization incrementally service-enables business functionality.

Gen2SOA Program Overview

1. **Browser Access to Legacy:** Within a portal based approach to SOA, a browser-based user interface (UI) can be added to a Block Mode/3270 or GUI application using CA Gen's Java Web Generation capability. This option is the most straightforward because it produces one web page for each existing screen, window or dialog box. This option is most applicable when little or no change in application functionality is required for integration into a service based web portal.

Everware-CBDI is working with a major financial institution to extend CA Gen applications to the web without reengineering. Through the use of Gen's Java Web Generation facility, Everware-CBDI is providing internal and external users the capability to directly execute Gen transactions from their web-based browsers. This effort leverages a significant investment in CA Gen capability, exposing proven business functionality to an extended user community through the net. The use of the Java Web Generation capability is allowing nearly 100% reuse of the server-side application, thereby significantly reducing development cost and time to market.

2. **Gen Proxies:** Gen Java, COM, or .Net Proxies can be used to provide the connectivity between the existing Gen systems and an implementation of an Enterprise Services Bus (ESB). The resulting services contribute to the SOA and can be consumed by other applications. This approach may involve some refactoring of the existing transactions depending on the current transaction structure.

Everware-CBDI used CA's Java Proxies to rapidly expose Gen server procedures to JSP/HTML clients for a State government agency. The use of proxies, generated from the CA Gen model, combined with a rapid prototyping approach facilitated the identification of business rules and an understanding of the required workflow, allowing 80 procedure/UI combinations to be developed in 20 days.

3. **Co-habitation:** This strategy provides a tactical approach to an eventual replatforming of the existing services within the Gen application. It starts out with a similar approach to the proxies, however instead of simple re-factoring; components are harvested

Gen2SOA Program Overview

within the Gen environment using CBD techniques. The result of this strategy is that the services and composite applications are separated within the Gen environment, essentially making the application service-oriented within the Gen environment. Once this is achieved, the process of migrating the application or any of the services (harvested components) is relatively simple since the application is already a modular service-based application.

Although clearly the most advanced strategy, Co-habitation in a SOA environment provides the ultimate in extensibility by laying a foundation for a full plug-and-play environment in which service-based components can easily be swapped in and out to meet the organization's changing requirements without respect to platform. Componentizing in this way sets the stage for establishing a complete service oriented architecture for application deployment.

Everware-CBDI has designed a services-based architecture for a major regional grocer that leverages their substantial investment in Gen applications. In the next phase of the project, service-based components will be mined from the Gen applications and exposed as services through the use of Gen's Java Proxies and JEE. The packaged Gen component services will then be consumed by other applications using an Enterprise Application Integration (EAI) infrastructure.

Various combinations of these strategies can be used to rapidly create services and composite applications that leverage existing Gen assets, while at the same time developing a comprehensive, iterative, low-risk approach for larger scale migration. And, of course, new business requirements can be implemented as a part of a **Gen2SOA** project.

Getting Started

There are many ways to begin a **Gen2SOA** initiative. Often a Pathfinder project is the best approach. The Pathfinder performs the upfront planning, accomplishes the training of in-house staff, identifies the major risks and architectural requirements of the overall project, and produces a deliverable that can be implemented and used by the business. Regardless of the approach, the initial step must produce an executable plan to accomplish the overall objectives of the project, such as:

- Speed to market
- Quicker ROI with reduced cost to service-enable existing Gen transactions
- Reduced project risk
- Controlled transition to service-based technologies
- Prolonging the useful life of existing infrastructure and systems
- Leveraging current skill sets in the migration to SOA

A number of considerations must be taken into account to determine the best **Gen2SOA** approach. The resulting approach will provide a basis to manage project risk and position the project for success. The following presents a sample of the issues that must be addressed:

1. Organizational

- Does an organizational policy, or statement of architectural direction, exist that will guide this initiative?
- Has the budget been approved to support this direction?
- Have the roles & responsibilities been defined consistent with the organizational direction? Is a training program in place?

2. Technological

- Is the technical infrastructure in place to support the direction or does a plan for it exist? Addressing such items as:
 - Quantity and type of server platforms
 - Application Servers (Java) and Web Servers
 - Network and Security
 - Backup and Recovery
 - Database Management System(s)
 - Enterprise Service enablement

- What are the characteristics of the anticipated service usage (volume, user type & experience, UI types, database activity, etc.)?
3. Processes
- Are the supporting functions defined and staffed to support the expanded user base the services will create through reuse? Such as, customer service, help desk, etc.
 - Has the software development process been updated to support a service oriented architecture?
4. Application Development
- What are the immediate and long-term objectives for the project and related subsequent projects?
 - How different are the service requirements from the current transactions embedded in the legacy systems and how volatile are they?
 - What are the current and future plans for development using CA Gen? What about existing Gen applications?
5. Current Systems
- What is the quality of legacy assets (their structural integrity, maintainability, etc.)? Are there plans to sunset these applications?
 - Does significant corporate knowledge reside in the legacy systems that would be difficult to replicate?
6. Strategies and Architectures
- Does a strategy document currently exist that governs IT decisions? If so, how forward looking is it (planning horizon)?
 - Have the information, application, and service oriented architectures (and network topology) been designed and developed for the future environment?
7. Business/Users
- Has business user commitment to the future IT direction been obtained? How are business requirements translated into IT priorities?
 - What business drivers exist that would impact the staging of transition releases?

Answering these and other readiness questions is an important step in developing an effective migration/transition plan towards SOA that will best

leverage the Gen legacy assets, as well as achieving corporate objectives, while gaining a better understanding of risk factors and timeframes.

Summary

The relatively recent proliferation of service oriented technologies has created new business opportunities and imperatives. To be effective, organizations must quickly respond by implementing robust, integrated solutions that capitalize on existing assets, provide high levels of user satisfaction, are quick to implement, and have a low risk profile. Everware-CBDI's **Gen2SOA** helps organizations rapidly integrate legacy applications with new capabilities on the integrating architecture of choice: Service Oriented Architecture (SOA). **Gen2SOA**'s flexibility in approach coupled with our deep experience assisting customers in successfully completing their migrations from Gen to SOA means that **Gen2SOA** can work for you!

Other Everware-CBDI Legacy Modernization Related Services ...

- **GenTran** – a program specifically designed to address the transition of software assets from Gen to Java or .Net. **GenTran** consists of development processes, methods, and consulting services designed to reproduce the existing Gen functionality to the flexible, open, widely supported Java/Java or .Net environment. The end result is a standards-based Java, Java, or .Net implementation for the business functionality previously found in the Gen application.
- **Java / .Net Express** – a short, focused project intended to exercise the Java/.Net environment and provide knowledge transfer to in-house staff while demonstrating the rapid delivery potential of service-oriented components by building a small, yet usable, application targeted to the business.