Building the Bridge to SOA

How does an organization reap the benefits of a service-oriented architecture while retaining the performance and value of its custom-built core systems?

By Jim Fowler | Software AG

Business White Paper

CONTENTS

The analogy of a bridge has to be one of the most common in human communication. Even a child soon figures out that a bridge is how to get from a familiar place to an unfamiliar place by crossing some formidable obstacle such as a valley or river. And the bridge always seems to look medieval, like something a troll would live under. That's a pretty boring and simplistic device for explaining technology. But what if instead the analogy was based on the actual components of a modern transportation bridge – and what if the bridge was a real bridge in a specific location? That would be much more interesting.

This paper is based on the five major components of the Sanibel Causeway Bridge, connecting Sanibel Island (off the southern tip of Florida) with the city of Fort Meyers on the Florida mainland (http://www.sanibelcauseway.com).

Hopefully these five bridge components will help create a vivid mental picture of the five components an organization needs to expose the unique value embedded in their core enterprise systems and extend that value into a modern service-oriented architecture (SOA). What is an SOA, and why is it desirable?

Although SOA is a complicated subject, the basic premise is fairly simple. The idea is to avoid building new applications from scratch each time, and also to avoid duplicating applications, data and business rules in one part of the organization that already exist in another part of the organization. Instead, applications are built quickly out of existing blocks (like a modular home), and draw information from different places throughout the organization without actually changing where the information "lives".

The way this happens is that useful "packages" of data and business rules (called *services*) are "exposed" as unique modules from a core system, ready to be used or "consumed" by a new business application or process. These services can be re-used over and over in different applications, and can be strung together with other services in a process called *service orchestration.* Re-use of services and service orchestration keep applications from being built each time from the ground up. The results of a properly implemented SOA are (among other things) greater agility, better responsiveness to customer and market demands, and significant cost reduction.

BRIDGE BUILDING 101

Before the five major components of the Sanibel Causeway Bridge can be used as an analogy for the Bridge to SOA, it is important to understand some physical bridge-building basics. The bridge is divided into two main parts, the substructure and the superstructure. The substructure is composed of the piles, footing and pier. The superstructure is composed of the beams and the traffic deck. All components are carefully designed to work together as a unit supporting the anticipated traffic load while withstanding all possible environmental factors (Florida = hurricanes). See Figure 1, below.

Superstructure

Deck: The top surface of the bridge superstructure that carries the traffic loads.

Beam: The main horizontal structural members that support the vertical load (in this case, the traffic deck). Beams are sometimes called girders. In the case of the Sanibel Causeway Bridge, the beams are pre-cast concrete.

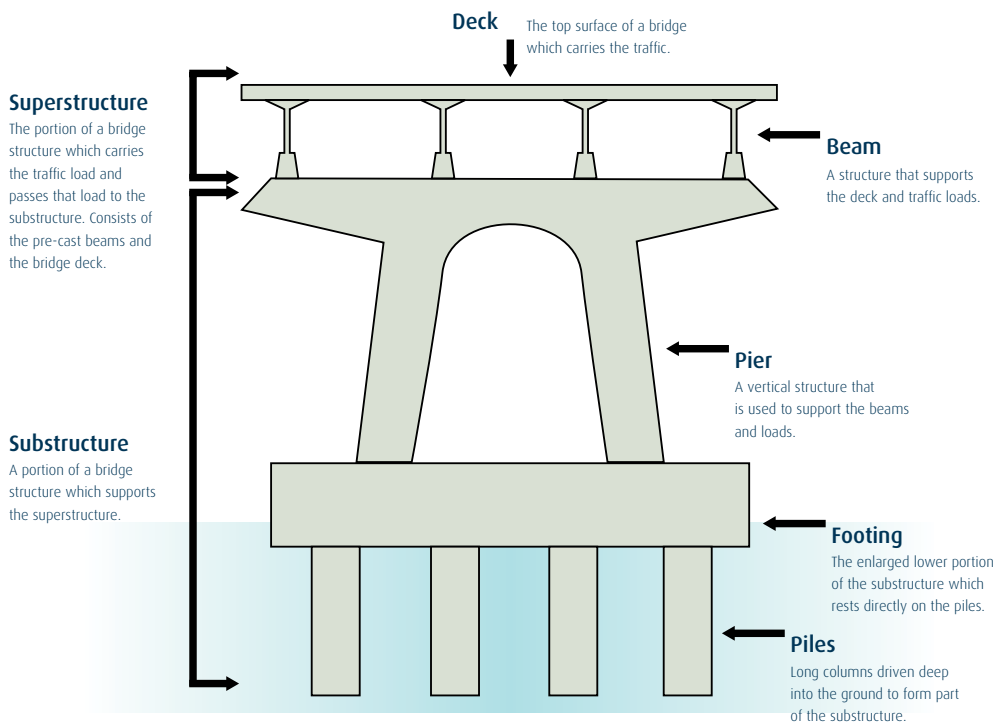
Substructure

Pier: The vertical part of the substructure used to support the beam and deck. It is the most prominent element standing above the water.

Footing: Enlarged lower portion of the substructure that rests directly on the piles. The footing and the piles bear the vertical load from the pier.

Piles: Long columns driven deep into the ground to form part of the substructure. They are driven by pile drivers until they reach solid rock or cannot go down any further.

Figure 1. Main components of the Sanibel Causeway Bridge, Sanibel Island, Florida USA

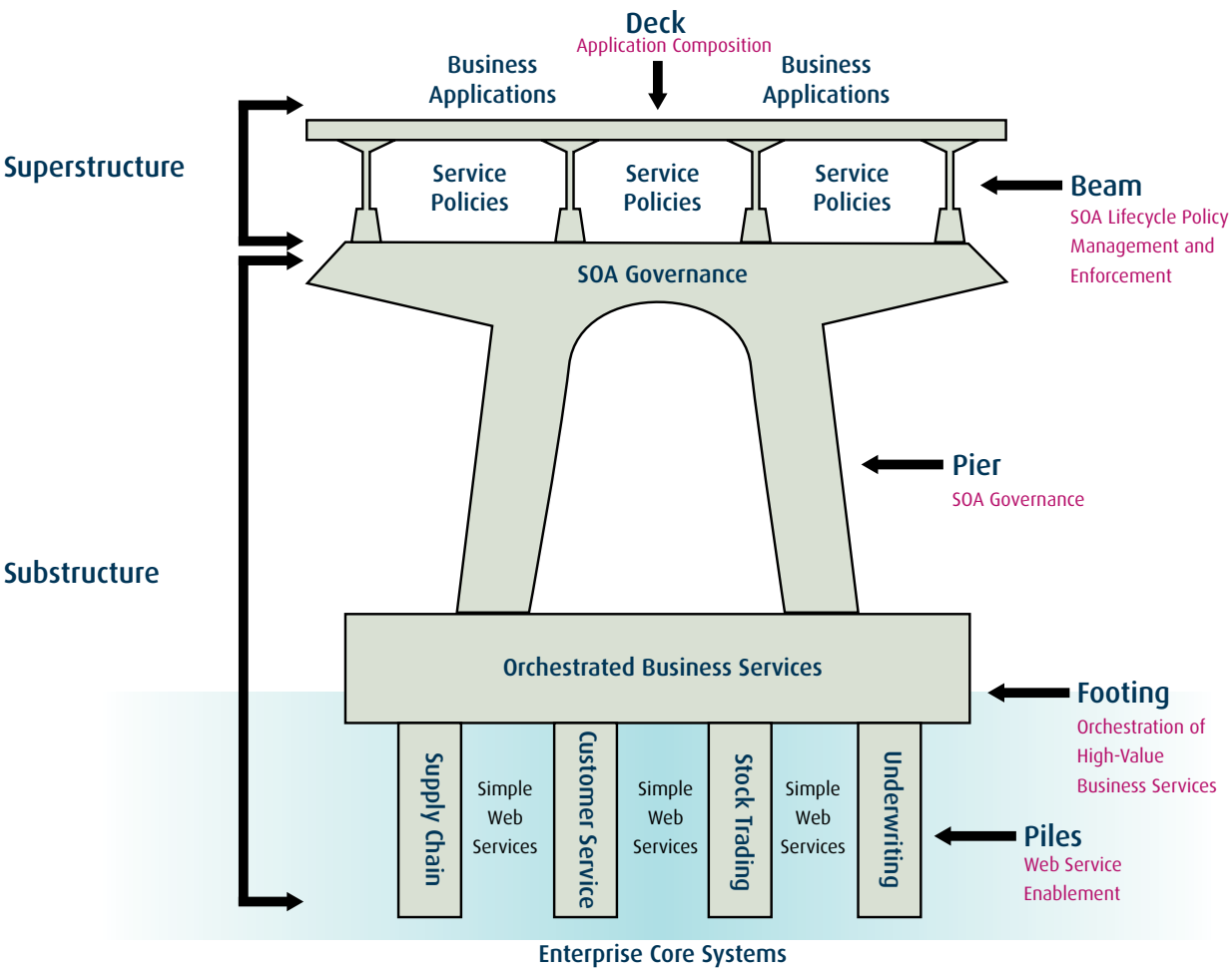


BUILDING THE BRIDGE TO SOA

The five primary components of the Sanibel Causeway Bridge map to five key elements that make up the Bridge to SOA. These elements are introduced below and in Figure 2. The remainder of the paper expands on the five elements of the Bridge to SOA and explains why each is critical to deriving new business benefit from existing core systems in the context of a service-oriented architecture.

A couple of elements of the bridge analogy are worth special comment. First, the core enterprise systems form the deepest foundation of the bridge, and each of the piles driven down into the foundation signifies a core enterprise system that is essential to the success of the business. Examples of core enterprise systems are a supply chain system in the retail industry, a customer service system in manufacturing, a stock trading system for large financial institutions, and a policy underwriting system in the insurance industry. In each case, these core systems, often residing on a mainframe, provide competitive differentiation for the business that depends on them.

Figure 2. Key Elements of the Bridge to SOA



These highly-customized systems cannot be purchased off-the-shelf, and no business would want their competitors to have the same system. Therefore, the service-enablement of these systems propagates unique business value all the way to the top of the organization – just as the piles of the bridge provide rock-solid stability to the entire structure. Without the piles going into the bedrock, the bridge would eventually shift around and become unstable.

Second, the analogy of the deck is very appropriate. Just as the deck of the Sanibel Causeway Bridge delivers the "service value" of physical support to each of the vehicles that cross the bridge, so the deck of the Bridge to SOA delivers the value embodied in each of the business services directly to the business users. In both cases, the deck is the interface. Furthermore, just as the deck of the physical bridge handles many kinds of vehicles (cars, trucks, trailers, campers, etc), so the highest level of the Bridge to SOA allows services to be used in many ways. In the case of this particular illustration, services are used in building composite applications. Important to note, services can also be used as components in a Business Process Management implementation or other piece of technology that directly serves the business.

These are the five elements of the Bridge to SOA:

Deck – Application Composition: Assemble new applications from multiple business services with little or no coding required

Beams – SOA Policy Management and Enforcement: Define, manage and enforce service policies across the entire lifecycle

Pier – SOA Governance: Facilitate maximum re-use of Web services within and across organizations

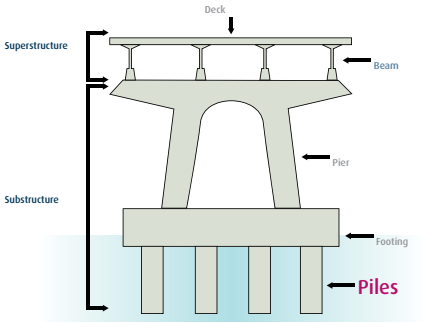
Footing – High-Value Business Services: Enable Web services to be "strung together" or orchestrated to create new high-value business services that provide greater alignment of the business with IT

Piles – Service Enablement: Extend valuable core applications to SOA environments without extensive programming effort

"DRILLING DOWN" ON THE BRIDGE COMPONENTS

When building the bridge to SOA, it is a good idea to start with the familiar, which is the part that is already in place in most organizations. The data and business rules that make up *core enterprise systems* (often running on a mainframe) help to differentiate the organization and make the organization successful. The idea of a "core" is very important. Geometrically, a core is the center of an object. Geologically, a *core sample* is what an engineer takes out of the earth by drilling down a significant distance – just as the *piles* of the bridge are sunk down a great distance into the sea bed and form the deepest foundation of the bridge. So the piles, analogous to service-enabling an organization's core systems, are the starting point upon which everything else is built.

When building the bridge to SOA, it is a good idea to start with the familiar...



The Piles: Service Enablement Overview

In order to re-use core mainframe assets in future applications, they must be exposed as services at various levels of *granularity.* The concept of service granularity is very important to SOA, and has to do with the level of complexity of the function. A service of coarse granularity performs a complex function. A service of fine granularity performs a simple function (Just as a core sample taken physically from the sea bed is going to have a mixture of coarse pebbles and fine sand particles, core enterprise systems will be exposed as a combination of coarse-grained and fine-grained services.)

The process of exposing core applications as re-usable services with differing levels of granularity is called *service enablement*. There are three basic ways to do service enablement: *session* integration, *transaction* integration and *data* integration.

The Piles 1: Session Integration

Many core mainframe applications are only accessible through terminal data streams, typically referred to as "green screen" terminals. This implies that the core applications are written in such a way that business rules and data access interfaces are not cleanly separated from the part of the program that can be viewed on a green screen terminal – the *presentation* layer. (The stream of information that puts characters on the screen in real-time is called a session.)

Separating the business rules and data access from the sessions in order to achieve service enablement can be expensive, time consuming and/or risky for some organizations. *Session integration* intercepts the information passed back and forth between the mainframe and the terminal, and is able to convert that information into HTML that can be displayed on a Web browser – or – the session information can be converted into services that can be re-used by other applications. Here are some pros and cons of session integration as a means of service enablement.

Session Integration – Pros and Cons

PRO	Completely non-invasive approach
PRO	No need to separate data, business logic and interface logic
PRO	Low risk, time and cost
PRO	Good when system is poorly understood
CON	Only "one-way" – the mainframe cannot consume services
CON	Cannot add any new functionality to the application – what you see is what you get
CON	Only works with online applications, not batch processes

The Piles 2: Transaction Integration

In some cases, core enterprise applications may be well-structured, with distinct layers for data access, business rules and presentation. This situation presents another possibility for service enablement to take place. The opportunity in this case is that the *transactions* (typically interactions with a database and inclusive of the business rules for manipulating the data) can be exposed as services. This is a similar process to session integration (above), in which the streams of characters that put visible images on a green screen terminal are exposed as services.

The technique in transaction integration is for a piece of software to put a "wrapper" around a transaction so that the transaction takes on the modular characteristics of a service. It is similar to the way packets of data are sent back and forth between devices in a communications network.

If a core system is not currently well-structured, yet it will continue to be developed and extended with new functionality, then it is probably worth the investment to reengineer that application to allow for transaction integration instead of session integration. This ensures a flexible environment to quickly meet the variety of future requirements that may arise. There are tools that will aid you in the reengineering process – by providing an in-depth understanding of application logic, interdependencies, and external touch points.

A successful transaction integration solution must be able to access many types of transactions, regardless of their language (Natural, COBOL, PL/1, etc.) or mode of operation (online or batch). It is equally critical for this solution to allow the application components to participate in an SOA without introducing change or risk into the environment. Therefore, these core system transactions need to be "wrapped" in such a way that they can be exposed as services without disrupting the original application in any way.

... core system transactions need to be "wrapped" in such a way that they can be exposed as services without disrupting the original application in any way.

Transaction Integration – Pros and Cons

PRO	Relatively non-invasive approach
PRO	Converts data types (Java <=> COBOL) in both directions
PRO	Works in online or batch modes
PRO	Offers the possibility of two-way communication – the mainframe can both expose and consume services
CON	Works only if application layers are well-defined, or re-engineering might be required
CON	The granularity of services might be too fine to be of much value in an SOA
CON	Enables access to data through the applications only – sometimes resulting in too much overhead

The Piles 3: Data Integration

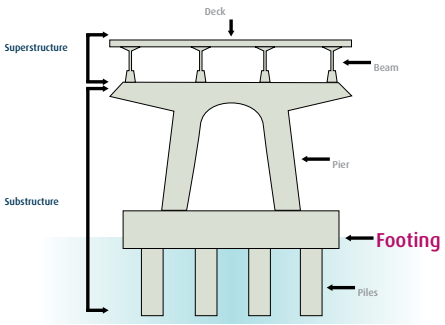
In certain cases, organizations need to gain direct access to data residing in mainframe databases, bypassing the presentation layer and the business rules layer. This is called *data integration,* and it is the third approach to service enablement.

In this case, data access logic is used to create a Web Services adapter. The new data access service can then be called from another application – perhaps to support creation of a composite application, corporate portal, or even to provide data to a business intelligence or reporting tool.

Data access services also provide a great approach for generating reports. Even if the same data is available through an existing application screen, executing a user interface service hundreds or even thousands of times to produce a report isn't the best approach.
to make use of the service enabled data.

Data Integration – Pros and Cons

PRO	Provides direct access to data rather than going through business rules (logic) to access data
PRO	Great for generating reports
PRO	Supported by many tools
CON	If a Web Service is used to update multiple mainframe databases apart from the original application's business rules, data integrity cannot be assured
CON	Data may be stored in a format unintelligible outside the context of the application built to access it
CON	Accessing data directly may imply that business rules are replicated on different platforms – going against the basic intent of SOA



The Footing: High-Value Business Service Orchestration

Once core enterprise systems have been service enabled, it is normal to discover that these Web services introduce additional complexities into the IT environment. For example, it was mentioned earlier that Web services created by "wrappering" core system transactions may be too fine-grained (performing a very simple function) to be of much value to other applications in the enterprise. Yet for a variety of reasons, creation of these fine-grained services may have been unavoidable, or even the best choice.

What is the proper course of action when a number of services are too fine-grained to serve the needs of the end applications or processes through an SOA? In such cases, it is often necessary to execute multiple fine-grained Web services in sequential steps, with additional business logic inserted between the steps. In doing this, a new, more "coarse-grained" service is created that is much more useful to a variety of business applications. This stringing-together of fine-grained services is called *service orchestration,* and a composite service created from such a process is called an *orchestrated service* or *high value business service.*

Typically, the piece of technology used to perform service orchestration is an Enterprise Service Bus (ESB). Service orchestration is not the only function performed by a full-featured ESB in conjunction with a service-oriented architecture. The ESB also performs high-speed messaging, routing and protocol conversion between systems, as well as supports various levels of security. This means you can be assured your Service Level Agreements (SLAs) or other performance expectations will be met.

Service enablement and service orchestration are great tools for building a flexible and reusable IT architecture.

The Pier: SOA Governance – The Challenge

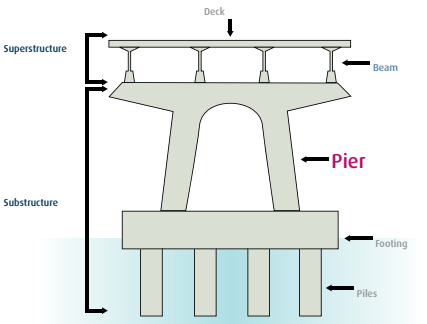
Service enablement (the piles) and service orchestration (the footing), the first two components of the bridge to SOA, are great tools for building a flexible and reusable IT architecture that enables a business to react quickly to new market conditions and customer expectations. However, when the adoption of services expands to dozens, hundreds, or even thousands of services in the organization, a number of new challenges appear. With many fine-grained and coarse-grained services being created and reused, the organization must be able to track information about the services such as:

- Which services exist within the organization?
- Which service is the most recent version?
- What are the access rights or security settings for a service?
- Who are the consumers of a particular service?
- How will a change to a services contract impact various systems and applications?
- Where are newly-created services published, and how are they documented?

These details and others will have a direct impact on the reusability of existing services, and on the success of the entire architecture. For this reason, it is now universally recommended by industry analysts that organizations adopt a method of *SOA Governance* as a central element of any service-oriented architecture initiative, and that governance be implemented right from the beginning.

The Pier: SOA Governance – The Solution

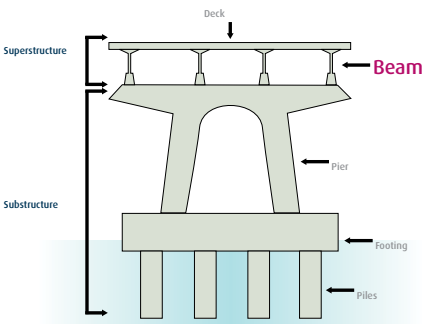
The obvious question is whether a single technology or set of tools exists that can enable the process of SOA Governance. Analysts again agree that a full-featured *SOA registry/repository* can provide excellent support for a program of SOA Governance. A registry/repository provides standard interfaces so that those who produce services can publish them and allow others in the organization to find those services and reuse them. A registry/repository also allows service producers to attach *service contracts* to their services – stipulating usage rights, security settings and other important parameters that must be respected by anyone who wants to re-use the services.



Consumers can be assured that whenever they bind or connect to a service, they will do so with the latest service contract. Likewise, service producers gain the ability to track how their services are used and by whom. The resulting implementation of the services registry/repository greatly increases the communication between the service producer and consumers, as well as the development teams. It is the central mechanism from which the various development teams can obtain the latest information regarding the service they need.

The Beams: SOA Policy Management and Enforcement

Service enablement brings the value of core applications forward into a service-oriented architecture. Service orchestration, through an ESB, creates the kinds of coarse-grained services that are most useful to the business, and are most likely to be re-used in a variety of new applications. SOA Governance, through a registry/repository, brings vital control to the process of service re-use, facilitating positive relationships between service creators and service consumers throughout the organization. However, the Bridge to SOA has not yet reached its final destination.



Once services are exposed and made available for consumption by various applications, they are only at the beginning of the SOA lifecycle. Each service will take on a life of its own as it is consumed in various applications, and as changes are made (or not made) to the system(s) from which the service calls data and business rules. All of the changing variables throughout the life of the service have the potential to wreak havoc within the enterprise infrastructure unless various *SOA policies* are created, managed and enforced. These policies serve to regulate each service throughout its lifespan.

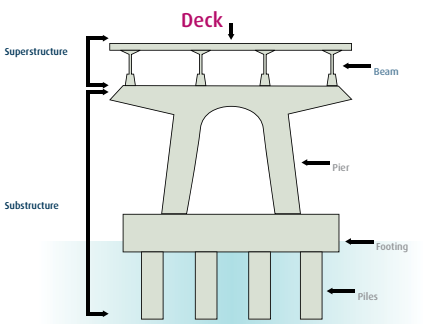
Governing the SOA lifecycle requires specific technology - a scalable SOA policy management platform that can span design, runtime and change time governance requirements. This platform sometimes also includes the registry/repository, combining all aspects of SOA governance under a single "umbrella."

One upside of an SOA policy management platform is that it can accelerate SOA adoption by making it easy to synchronize policy enforcement across the SOA lifecycle.

One upside of an SOA policy management platform is that it can accelerate SOA adoption by making it easy to synchronize policy enforcement across the SOA lifecycle. Another upside is that SOA policy enforcement technology can enable organizations to consistently meet their Service Level Agreements (SLAs) and Key Performance Indicators (KPIs) and automate the SOA processes that support their business objectives. Here are some of the specific functions of an SOA policy management platform:

- Guide SOA Adoption – support and enforce advanced SOA governance polices across multiple SOA stakeholders
- Connect the lifecycle processes – enable diverse stakeholder groups to collaborate, approve, and be notified of lifecycle events associated with policies and services
- Automate and synchronize policy enforcement across the SOA lifecycle - ensure consistent event-driven outcomes across the entire organization as well as across multiple organizations and B2B relationships
- Enforce architectural standards – automatically validate new services for policy compliance and established approval processes
- Enforce security procedures – set parameters for various levels of access to information about services
- Protect confidential information from misuse, while allowing access to authorized users.
- Extend governance ROI to external partners – Internet-enable products and services via governed Web services

While service enablement and orchestration came about fairly early in the SOA movement, the need for sweeping policy management and enforcement only became abundantly clear when SOA entered widespread adoption by organizations worldwide (although many IT analysts will surely claim foresight). It seems that the early adopters of SOA had little idea how much genuine chaos would be created within the enterprise by the phenomenal success of the methodology they were pioneering. SOA governance – including registry/repository and policy management and enforcement – can now help IT organizations ensure that there is never "too much of a good thing".



The Deck: Application Composition

This paper has often referred to "new applications" that are the consumers of the services exposed from core systems. Clearly these applications, and the business goals they serve, are one of the primary reasons to build an SOA in the first place. In the Bridge to SOA example, they are the paved surface over which all the meaningful business traffic moves. But what are they?

These new applications are the result of a process called *application composition,* and are known as *composite applications*. They are not developed in the traditional manner, by writing line after line of code in a language such as Java or .NET or COBOL. Instead, composite applications are created in a modular way by combining Web services available both inside and outside the enterprise.

Composite applications are typically accessed by the business user on a Web browser similar to Google or Yahoo (in fact, Google Maps is a popular component of many composite applications found on the Web and in business environments). Yet these applications must also have enterprise-level speed, performance and graphical interaction capabilities associated with sophisticated desktop applications.

Efficient, "codeless" assembly of composite applications for business users requires an enterprise-class Web application development environment. Such a development environment will often have impressive built-in functions. These include...

- The ability to quickly incorporate "Web 2.0" features into applications such as RSS feeds, Google Maps, threaded discussions, Wikis and instant messaging
- Sophisticated user management, role-based security, and document management capabilities
- The ability to deliver Web-based applications without requiring new software to be installed and maintained on end user machines, ensuring a lower total cost of ownership
- The ability to create different user interfaces and page layouts for different users, groups, or roles by mapping specific users, groups, and roles to different page layout elements
- The ability to assemble new applications that were never dreamed of before. (When some of the mainframe applications were first developed, did the developers ever think that someone would be able to access their systems from a mobile phone?)

Application Composition completes the Bridge to SOA because the user of composite applications will actually be using services to call vital information out of the organization's core systems. Previously a business user might only have accessed this information in isolation by requesting a report from IT or sitting in front of a special green-screen terminal. Now they can receive the information transparently, in a graphical browser on their own computer, in a context that makes sense for the particular business problem they are trying to solve. This is truly revolutionary.

SUMMARY – TRANSPARENT BUSINESS SUPPORT

When automobile drivers traverse a busy traffic bridge such as the Sanibel Island Causeway Bridge, they probably don't think about the discrete bridge components that make their quick trip over the water possible. Instead, they think about the vacation they're about to experience, or the people they will spend time with. The Bridge to SOA is the same way. Business users will enjoy the speed, convenience and flexibility they possess with their new browser-based applications. But the service enablement, service composition, SOA governance, policy management and application composition capabilities that support the applications will be completely transparent – just like the reliable mainframe core systems that still provide value and never seem to go down.

TIP #1: Drive test piles before starting the job.

SOA Equivalent – Before starting to Web-enable core applications, determine the proper granularity of the Web service. Too fine a granularity and services will be meaningless to a business user. Too coarse a granularity and the services will not be re-usable.

TIP #2: When designing the bridge, be fully aware of all environmental factors that could stress the components.

SOA Equivalent – Be sure there is full buy-in on the "bridge to SOA" project by all stakeholders, including IT management, enterprise architects, business line managers and key executives.

TIP #3: Design the bridge in anticipation of the expected traffic load that the deck must carry in the years ahead, not just the current traffic load.

SOA Equivalent – Don't scrimp on SOA Governance. Perhaps the current number of services can be tracked on a simple spreadsheet, but the infrastructure should accommodate a "wildly successful" process that is adopted throughout the organization

TIP #4: Don't try to save initial cost by building the bridge with inferior materials; this invites disaster down the road.

SOA Equivalent – a project that will impact the future of the business should be entrusted to partners proven to offer world-class experience and mission-critical software solutions. Building the Bridge to SOA is not a task for small niche vendors or quick-and-dirty methods.

TO FIND THE SOFTWARE AG OFFICE NEAREST YOU,

PLEASE VISIT WWW.SOFTWAREAG.COM

Take the next step to get there – faster.

ABOUT SOFTWARE AG

Software AG is the world's largest independent provider of Business Infrastructure Software. Our 4000 global enterprise customers achieve business results faster by modernizing, integrating and automating their IT systems and processes. As a result, they rapidly build measurable business value and meet changing business demands. Based on our solutions, organizations are able to liberate and govern their data, systems, applications, processes and services – achieving new levels of business flexibility.

Our leading product portfolio includes solutions for high performance data management, developing and modernizing applications, enabling service oriented architecture, and improving business processes. By combining our technology with industry expertise and best practices experience, our customers improve and differentiate their businesses – faster.

Software AG – Get There Faster. ™

SAG_BridgeSOA_WP_13May08