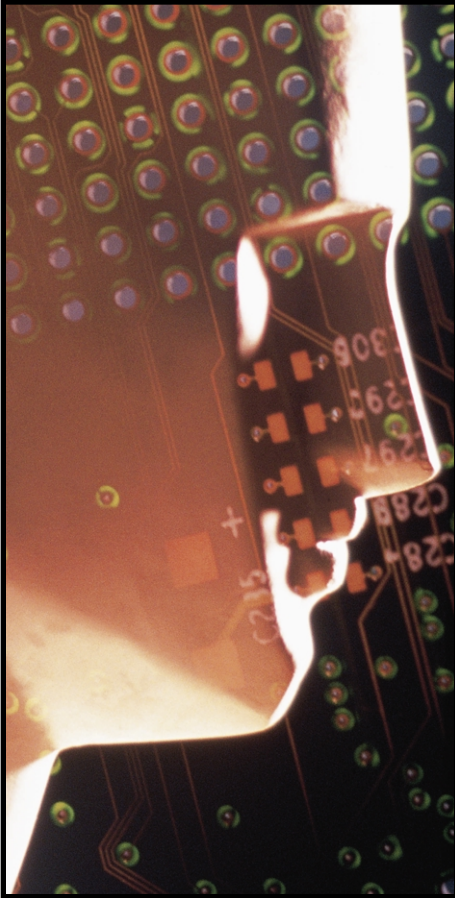


Service-Oriented Architecture

September, 2005



SOA: A Business Architecture for Managing Uncertainty

One of the most ironic contradictions in the way we respond to an increase in uncertainty and risk is the natural tendency of most people and businesses to slow down decisions. Additional time is typically spent evaluating alternatives and scrutinizing threats. In short, every decision gets dragged out to the last possible moment. The irony is that uncertainty actually decreases the window of opportunity within which to make decisions.

Good decision making requires more than good instincts and a solid gut; it requires information, management and business systems that keep pace with the velocity of change and uncertainty in the world around us. Only in this way can we mitigate risk and seize opportunities before they expire. Yet most organizations rely on information systems that are hardwired to deal with anticipated problems and opportunities, rather than to deliver on-demand services, products, and solutions that are best suited for the precise requirements of the moment.



A Perot Systems Company

111 Huntington Avenue, Suite 2750
Boston, MA 02199

Evolution of Service-Oriented Architecture

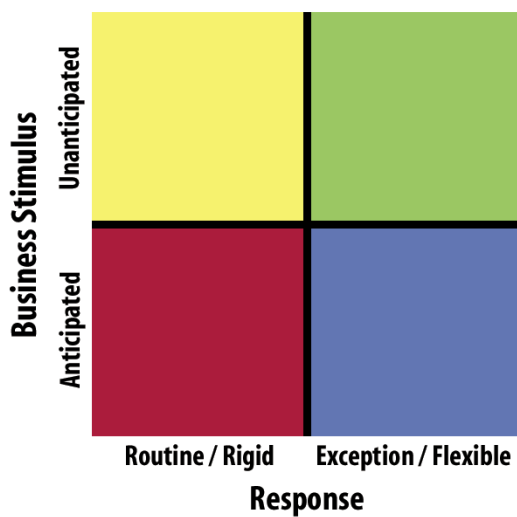
So how does an enterprise break away from the rigid legacy of these systems to achieve the levels of agility and responsiveness necessary to succeed?

To Start With, You Need to Accept a Basic Tenet

Becoming exceptional at your business means becoming extraordinarily adept at handling exceptions.

Managing exceptions has become a core requirement for successful organizations – it is the unanticipated that we have to prepare our organizations for. The most important, opportune, and riskiest transactions are not the routine or unexceptional.

One way to visualize this is to relate the demands of the market to the responsiveness of your organization. Market demands can come in two forms: routine (markets ask for products and services that you know how to produce); and spontaneous (markets ask for products and services that you have not yet produced). At the same time an organization can respond to both of these categories of market demands in one of two ways: a predetermined fashion or an innovative fashion. The information systems of most organizations rely upon predetermined responses to anticipated demands.



These hardwired systems exist in the lower left hand quadrant of the framework shown in the diagram to the left. But what about the other three quadrants?

Addressing the market's demands outside of standard, rigid transaction-based systems, such as ERPs and CRMs that respond to anticipated stimuli using routine processes, requires creating an amalgam from systems that were just not developed or even contemplated to work in the same organization, much less the same

value chain – what’s needed is something far more flexible and dynamic than the monolithic, chiseled-in-stone solutions available to date.

This is where Service-Oriented Architecture has the potential to change not only how we build information systems but also the very nature of how we do business.

Simply put, a Service-Oriented Architecture (SOA) is an approach to building information systems that provide nearly instantaneous responsiveness to changing market conditions – closing the gap between threat and opportunity.

An SOA Relies Upon Some Very Basic Fundamentals

- *A component-based model for constructing applications on-demand – If we break down all applications to their most basic building blocks we have a library of business objects, each one performing a specific function. This is not unlike standardized parts for building electronics.*
- *A discovery process to allow you to find the component that you need at a specific moment in time, to work with a specific set of other components, to construct an on-demand application for a specific set of market circumstances.*
- *A registry that contains the vital information about every available component in order to describe basic capabilities and compatibility issues.*

It is only recently that platforms, network interoperability, and standardization have evolved to a point where they can support the vision of these sorts of distributed, component-based applications. But, perhaps more to the point, the current economic climate has forced a crisis of integration upon every organization. There is a tremendous flight to integrated solutions which can provide liquidity in the information and process assets organizations are creating.

The Foundation for SOA: Building Applications from Components

Componentization and its inherent benefits for software reuse has been long understood within the world of programming. Since the earliest days of programming developers have tried to reuse code in order to minimize the time it takes to create a software program. Reuse and componentization is hardly revolutionary. However, the ability to build libraries of components into a vast network of distributed services that can be reused by not only programmers but integrated on-demand by any application that has access to the network is a radical shift in how applications are built and how they evolve. In this new model of automation even the term network starts to fall short of adequately describing the incredibly granular level of these tiny components. A more appropriate metaphor would probably be that of molecules binding and separating in myriad chemical reactions to form new compounds. This is the core concept of software on demand or what is increasingly being referred to as a Service-Oriented Architecture (SOA).

While most of the attention focused on SOA or on demand has been largely dominated by Web services, it is important to realize that standards such as Web services are merely a means to an end and not the end game itself. The true value of SOA comes from the ability to orchestrate components across business processes.

By definition, the use of Web services presumes that automation is already in place (i.e., Web services enables connectivity between automated activities), yet very often this exists as the proverbial islands of automation, isolated from each other with no visibility between them. One system might handle technical support documents, while another coordinates call center schedules, and another tracks problem resolution status. Thus each system is responsible for a discrete function, yet lacks the ability to determine if or where a process may have broken down within any of the related but otherwise disconnected systems.

Creating a business that possesses the ability to handle uncertainty through agility, and liquidity will require both a practical set of current day solutions to the challenge of integration and a commitment to a new vision of how business and IT architecture work together.

The Next Generation of Software Architecture

While first and second generation integration technologies, such as ERP, CRM, EAI and portals, offered dramatic advances in the presentation and hardwired interoperability of enterprise and desktop applications, they were limited to very specific points of integration by proprietary *Application Programming Interfaces* (APIs) and were rarely able to integrate spontaneously with the critical touch points where customers and partners interact in complex, fast changing digital value chains.

This is where the interactions among the constituents of a value chain become least predictable and most risky. By orchestrating these interactions and providing a universally integratable platform for software, **SOA solutions promise to alter software development, enterprise applications and the value chain in a manner as radical as interchangeable parts altered manufacturing.**

The result will be the evolution of information systems which consist of componentized parts that work seamlessly across the entire digital value chain.

SOA may sound a bit far fetched; it shouldn't. Once every decade or so the market for enterprise class software solutions tilts dramatically causing dominant players and products to lose traction as they seemingly fall off the edge while new players and products rise to prominence.

Whether or not your business involves a call center or support desk, this is nonetheless likely to be a familiar scenario for any firm that has invested in automation without the orchestration of an SOA.

One approach to this problem would be to manually integrate these otherwise disparate systems with hardwired points of integration. The problem is that this sort of an approach would work for those situations that we can anticipate. Yet this would not address the need to rapidly coordinate processes in unanticipated situations. So what is the answer and why SOA?

SOA makes applications transparent by exposing their capabilities as objects. That sounds a bit obtuse but it is very straightforward and has precedent in many other industries. For example, you would expect to be able to use off the shelf components to build your home entertainment system, even if each component came from a separate manufacturer. Take that to the next level and, if you had the skills, you would also expect to be able to build the circuitry of an amplifier or tuner from basic components available at your local electronics outlet.

Similarly by breaking down large applications into small standardized component applications and making them available over the web, where they can be recombined in almost limitless combinations, we can start to create vast libraries of recombinable objects.

Most SOA discussions end there but disconnected objects alone would do little to help coordinate processes across large enterprise systems. The problem is that without a process-based add-on solution, for the objects to be “discovered” you have little chance of finding what you need quickly enough for it to help.

This represents a significant difference between SOA and prior approaches such as Enterprise Application Integration (EAI). It is also why you hear so much about the concept of a “registry” in an SOA approach. The registry is the clearinghouse for objects. In the same way that a job seeker might register with Monster.com, an SOA object

Our ability to quickly forget the way software platforms can change so dramatically does not serve us well in anticipating the future and seeing the next shift.

Consider this, who was the leading software only vendor in 1980? It was not Microsoft, Oracle, IBM, or SAP. It was a name long forgotten by most people and unknown to anyone under the age of 40 – Cullinane, which later changed its name to Cullinet and revolutionized the software industry with packaged enterprise (then known as departmental) applications.

These applications allowed enterprises to achieve greater levels of agility by not only automating many of the mundane transaction-oriented processes of accounting and administration but also allowing for exception processing and modification of these systems over time.

This trend continued from packaged applications, and RDBMS, to word processing and spreadsheets, to ERP and CRM. What each of these tipping points had in common is the degree to which they created technology platforms for businesses in the face of increasing uncertainty as to their practical application. In other words, none of these technologies was fully formed as a solution when they were first deployed. But when critical mass was reached each one became an immutable factor in the agility and success of a competitive business.

During this same time, however, applications underwent the equivalent of an enterprise land grab as each enterprise software vendor attempted to occupy as much enterprise IT real estate and mind share as possible. No wonder then, given this winner take all market mentality, that in the current economic and organizational context, the most frequent topic of discussion is that of integration.

must register its capabilities and pedigree in order to be found and evaluated. In SOA terms we say that the object (also called a service) is self-describing and self-contained – if you will, it has a resume of its capabilities.

Of course, the metaphor gets much more complex since changes within an application constructed from registered objects will need to keep track of changes to new versions of the objects and constantly reverify that they work together with other objects in order to avoid incompatibility. Anyone who has dealt with other forms of integration knows full well the problems involved with application version changes breaking integration connections. In the context of live processes, these changes will bring work to a screeching halt.

However, with a fully functional process for discovery and registration SOA becomes a very attractive option in the climate of fast changing markets, accelerated innovation, and extended digital value chains. Imagine that in short order, certainly within the decade, applications will quite literally disappear and be replaced with global utilities of information objects that can be brought together with near zero-latency to construct a specialized solution to address any market opportunity or threat – foreseen or unforeseen.

SOA is today certainly far from having proven this sort of large scale viability for component built applications, object discovery and registries but the appeal of components to organizations hungry for greater flexibility, lower cost of ownership, and ultimately faster responsiveness is clearly going to push the evolution of SOA at a rapid pace.

However, seemingly adequate solutions insulated from each other within the application stovepipes of an organization are now horribly inadequate for use within an integrated framework. **The scenario is not unlike taking best of class automobile parts from several automobile manufacturers and then trying to build a working car – which must then be rebuilt each time it crosses state lines in order to conform to local transportation codes.**

With this fragmentation across enterprises, digital value chains are anything but integrated. The vast majority of the glue that holds digital value chains together is made of elbow grease and manual intervention in the form of splintered processes that have come to define the landscape of information workers.

The Evolution of Services

During the last two decades enterprise applications have increased in numbers, complexity, size and cost. Despite the enormous reliance on enterprise applications, fragmentation has created fiefdoms and poor integration with little practical architecture for future integration. Instead the enterprise application landscape was an architecture built for competition and dominance.

However, while applications were evolving, so was infrastructure – that is the basic services shared by all of the piece-meal parts of an enterprise solution. As infrastructure grew in its importance and became a platform for business transactions, aided in large part by the Internet, it began to subsume applications.

The gravity holding together these integrated apps, however, is still hardwired, with myriad home grown patches providing the glue. Although the appearance of integration may have advanced significantly in that last few years through the application of portals, the underlying components are still discreet and mostly incompatible bits of code.

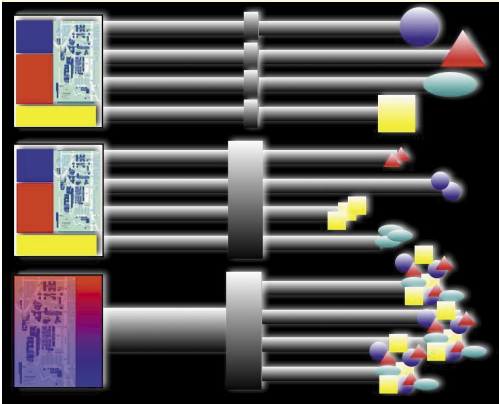
Three Eras of SOA

The move towards virtualizing information systems involves three distinct eras of architecture.

Enterprise Application Integration (EAI)

The first era was that of visualization. Large numbers of proprietary connectors were developed by application vendors which would communicate with or launch specific subsets of the application. The standardization in this phase was communication-based and primarily between the links from applications to components. You can envision this as a series of very thin pipes between the applications and an integration engine, each thin pipe developed to facilitate communication between a connector (aka API) and an enterprise application.

On the other side of the integration engine an equally large number of pipes would be used to pass the visually homogenized information to a user interface that combined the output of the pipes into a reasonably coherent single presentation.



Over time however the sheer volume of components, their proprietary nature, and the complexity of the visual representation of so much disjointed information working together, which it was never meant to do so, became unwieldy.

Web Services

The second era, which we are now in as an industry, has begun consolidating the chaos by inserting a set of standards between the integration engine and the applications. This

Fluid, Not Scripted

Although very attractive in concept the vision of an SOA strikes chords of fear for many software marketers since it represents an inherently ambiguous, and as result, difficult to quantify solution in terms of traditional software features or functions. It is also a model that spells increased risk in the vernacular of many CIOs who see it as increasing the number of possible points of failure in an IT architecture.

Whereas traditional transaction-based processes are necessarily scripted and defined when automated with software, an SOA is fluid and requires facilitation, not control. **The fluidity of collaboration, the liquidity of information – these are the new factors of competitive advantage.**

This is a subtle yet explosive notion that presents one of the greatest opportunities for disrupting the traditional notion of work, and is at the very heart of why SOA matters.

Still, the concept of an SOA is a catalyst, not an end unto itself. It is a reflection of the business environment we are in today, where org charts have become all but obviated, and competitive advantage is more often measured by the speed with which virtual teams can rally together. The post-dotcom era has brought results and execution back into the spotlight. Yet this should not be interpreted as a call to return to outmoded processes and practices. However often the “back to basics” mantra is repeated, we cannot turn back the clock. Work and organization as we knew it has forever changed.

begins with the standardization of the connectors themselves so that they can interoperate freely with each other. This transformation is key to the componentization of the connectors and is most frequently referred to as the shift to web services, since the components can now meld more easily into a single presentation. Standards such as WSRP and JSR 168 are essential for this.

In the second model componentization allows for greater adaptability in the way applications are constructed. However, the components now require increased levels of sophistication in their administration and maintenance. In addition, the components are still being discovered and integrated manually. It works for small implementations but becomes unruly and unreliable when attempted over multiple organizations.

SOA

This leads to the final phase, service virtualization. In the virtualized model, a single virtualized application is delivered to the user's point of access for every instance of use. The libraries of components have now become vast networks of distributed, discoverable, registered services that can be polled and enlisted on-demand.

This approach is the essence of an SOA, bringing together applications on-demand at the point of user access by dynamically binding the pieces of the solution. This shifts the field but does not yet tip it.

The tipping point and the key enabling aspect of an SOA is easy to miss but critical to its evolution – a Virtual Channel through which to broker large blueprints for solutions that end up taking the place of what we today call enterprise class applications. In the virtualized model services do not now flow directly from application or infrastructure to the user, as they might with an EAI approach, but rather through a virtual channel based entirely on user requirements. In this model service providers emerge to provide the security and reliability of large scale SOA integration. In many ways we see these providers as being the next wave of the information industry.

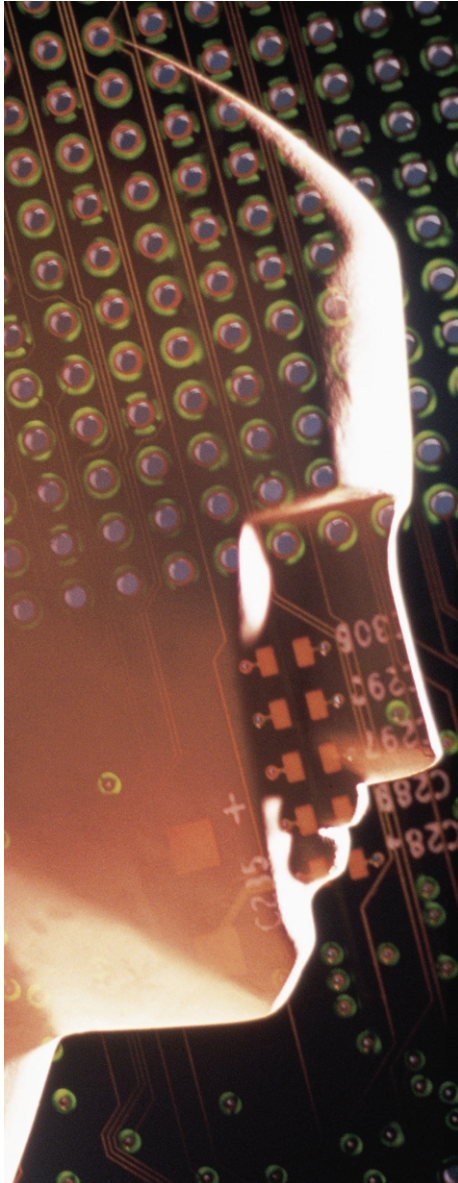
Looking Forward

If the idea of Service-Oriented Architecture is tough to buy into because of its complexity or the somewhat circuitous route the service takes to get to the user's point of access, consider the analogy of telecommunications. Telecommunications did not take off until it evolved similarly through the use of vast networks. If you had suggested to someone fifty years ago that they would take a route that may involve several thousand miles of network traversal to call someone only a few miles away they would have clearly laughed you out of the room. Yet that is precisely the way telecommunications and the Internet have evolved. Neither is a point to point metaphor but rather an intricately brokered model where even the most basic peer to peer communications are routed through very complex gateways and channels.

The practical reality is hard to ignore. We have entered a period of immense volatility and uncertainty that makes planning a daunting challenge. For our businesses and our economies to survive, the fundamental architecture of our businesses and our information technologies must evolve to keep pace.

In the same way that the foundations of a building remain intact and immutable while the structure may be remodeled many times, the foundations laid for information systems must support an ever malleable business structure, which can take the shape most appropriate to its environment.

SOA may still be more vision than reality but its mandate is a clear prescription for organizations that expect to remain viable and competitive in the decades to come.



**BPX
2005
SUMMIT**

*The Premiere Event for
Business Process Professionals
October 16-19, 2005*

Hotel del Coronado San Diego, California

**SOA
Summit**

*The First Executive Symposium on
Service-Oriented Architecture
October 19-21, 2005*



Produced by Delphi Group

About BPX 2005

BPX 2005 is the premiere event for business process professionals, now in its 8th year at the fabulous Hotel del Coronado. BPX features over 36 sessions designed to teach you how to pick your first process to manage and optimize, how to perform business process analysis, define business rules and business logic, as well as how to understand the BPM software landscape, staff projects, and build a business case to win sponsorship and get projects funded.

*For the Complete Conference Program Visit
www.BPX2005.com*

About the SOA Summit

The SOA Summit is a first-of-its-kind event, providing real-world validation of leading-edge SOA and Enterprise Information Architecture concepts and strategies, through an expert-designed curriculum of case study sessions and leading Chief Technology Officers and Chief Architects.

*For the Complete Conference Program Visit
www.SOASummit2005.com*



A Perot Systems Company

www.delphigroup.com

About Delphi Group

Delphi Group is a leading provider of business and technology advisory services to Global 2000 organizations. With offices established around the world, Delphi has assisted professionals across disciplines and industries at nearly every major national and global organization and branch of government. Its clients and subscribers include more than half of the Global 2000. All content ©2005 Delphi Group.