

High Performance Service Oriented Architecture - SOA: Part I

Service Oriented Architecture (a.k.a., SOA) deserves careful consideration because it is the first technology framework that delivers key ingredients requisite in streamlining complex and high-speed business processes. Business people have reasonably mature financial (a.k.a., income, balance and cash flow statements) and strategy frameworks (e.g., Kaplan's Strategy Maps). However, until recently organizations were missing a high fidelity IT process framework that could effectively link the 10,000's of its activities to the highest levels in the business. SOA can help deliver this capacity while reducing or eliminating the lack of accountability that often occurs in the "IT Twilight Zone" – that warped space between business and IT operations. Assuming SOA can deliver these capacities, three major areas of business operation would be well formulated – strategy, finance and information related operations. Print this out, fold it and stick it in your back pocket. Read it once, read it twice and maybe a third time. If it does not help you cut through the current confusion about the most important framework for thinking about IT, then please kick me!

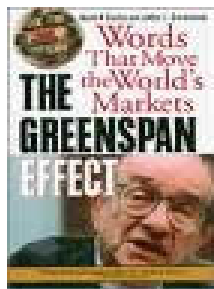
What's in Store? Distilling Fact from Science Fiction

Throughout the last 2 years, there has been an acceleration in the hype over SOA. The next few articles examine SOA, the Enterprise Service Bus (a.k.a., ESB) and how IBM WebSphere products support these conceptual frameworks. In this newsletter, we primarily discuss the business aspects of SOA and where the ESB fits in to the SOA scheme. I will attempt to distill SOA facts from fiction. An outline of IBM's solution for SOA and the ESB is shown. Two additional newsletters are in the SOA series and include:

1. **Part II** – Technical review of Enterprise Service Bus (ESB) concepts and IBM products capabilities.
 - WebSphere Enterprise Service Bus V6 (WESB)
 - WebSphere Message Broker V6 (WMB)
 - WebSphere Application Server V6 built-in Service Integration Bus (SIB)
2. **Part III** – Technical review of WebSphere Process Server V6.



Alan Greenspan – US Productivity Limited by the Nation's IQ



In a recent speech circa October 2006, Alan Greenspan, renowned US Federal Monetary Chairman had this to say about the limits to our economy and productivity (performance):

"Is productivity going to increase? The answer is, of course it is." But it's difficult to predict just how quickly. Moreover, productivity gains of more than 3% per year are nearly impossible to sustain over multiyear periods because businesses -- and the people that run them -- can only adapt just so quickly to exploit new technology."

"The obvious answer is that we're just not smart enough. How we apply cutting-edge technology is a function of IQ."

Mr. Greenspan also went on to say that there is a dramatic bottleneck in U.S. healthcare related to technophobic senior physicians.

*“The growing use of electronic medical records will help the healthcare industry better identify and implement best practices, Greenspan said. **But the industry won't fully embrace IT until younger tech-savvy physicians predominate, replacing technophobic physicians now in their 50s and 60s. And it's impossible to predict the impact of IT on healthcare costs. Medicare soon won't have the resources to provide healthcare to all the aging baby boomers. We have probably over promised to baby boomers what can be delivered.**”*

This implies that not only IQ is important to productivity but also other factors such as risk tolerance.

The Business of SOA – Conditions, Value Proposition, Risks

SOA as a concept as well as technologies that enable it are approaching maturity. Organizations that are prime candidates for SOA have complex processes that need to be fast and have the ability to change with fast moving business environments. The **value proposition** is significant because of the intended business focus while leveraging inherent economies of scale through service reuse across LOB's. Previous innovation cycles often focused more on IT – witness either distributed computing with CORBA or the application server revolution. However, there are **risks**. SOA requires tech savvy and disciplined management, substantial centralization of SOA planning and engineering, potentially a change to incent managers less on completing projects for LOB's and more for the enterprise's profitability. In the past, only the most IT savvy companies had mature software engineering process and the people talented enough to follow them. SOA requires even more engineering up front and without it, organizations are at risk of having an overwhelming failure with their initiatives. Overwhelming in the sense that SOA has enterprise wide implications. As with prior IT initiatives, SOA will be hard to quantify in terms of ROI. Several critical questions need to be addressed. One, who in the enterprise will pay the additional cost to build services for enterprise wide use. Two, once enabled, do other LOB's get services free? Third, will past-unprioritized access to resources prevail or will asset utilization finally be linked to business priorities and profitability.



What Conditions Make SOA Desirable?



Three key conditions make SOA desirable: One, the combined complexity of both communications between and within all the enterprises applications; Two, the need for speed in modifying processes and three, the need for pure processing speed.

1. **You have complex processes that are not well understood resulting in increased costs and lower throughput.** Complexity increases the cost of maintenance and process operations because complexity begs for expertise – expertise required for critical components in the value chain are often scarce and expensive. It also slows processes down because human intervention is required more frequently to referee complex decisions. Typically, larger companies have complex processes that should be better understood. Using SOA can be a good way to drive and sometimes push the organization to understand their processes better. Improved processes cannot only conserve cash, but become a competitive advantage. Small companies with simpler

processes and/or low complexity applications are likely not the best candidates for SOA. To be fair, defining what is complex is somewhat arbitrary.

2. **Process cannot be modified fast enough to keep up with changes in your business environment.** If the organization does not have to change its processes often then they may not be in need of SOA.
3. **Your customer demands speed or your industry is fast and super competitive.** For example, if FedEx promises next day delivery, then it has to be there the next day. This is an industry characteristic. A bakery has to get the bread out that morning because it is a characteristic of the product – otherwise it does not meet its freshness quality attribute.

What's the Value Proposition of SOA (a.k.a., ROI, NPV, CVA)?

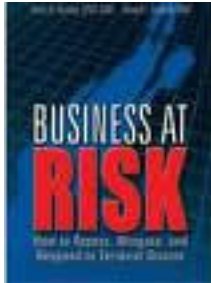
The valuable arguments in our view are SOA's inherently intense focus on the business, potential to leverage economies of scale through service reuse and the ability to act as a vehicle to drive out improved process knowledge that will lower cost while delivering faster process execution.



1. **A business focused framework first, technology as an enabler.** Past waves of innovation focused substantially on improving technology that often lost its direct traceability to business needs and subsequent ROI. Although SOA does not force a focus on the business, it does push implementers to think services closest to the business domain. Business versus technology domain services differs in that the business services correlates directly with business goals. For example, “Get Checking Account Balance” versus “Encrypt Data” would each be part of business and technology domain services respectively. To gain the maximum benefit, the services created need to be offered to the greatest number of LOB's to leverage economies of scale. For example, 25 LOB accessing an application in 25 different and non-standard ways slows down organizational change capacity while inducing excessive effort and cost in terms of development and maintenance duplication. The more effective way is to set up one service and have all applications access it through one service channel. To be fair, there is the question of how will the creation of one new service be negotiated amongst all those LOB managers? That is a good question and is addressed further on.
2. **Efficiency with scale – leverage the cost across many users.** In tech talk, this is called reuse. Economies of scale are important because it is a basic tenet of economic theory that can help maximize profit. For software products, adding the first user is very costly; in fact, the whole system cost. Adding the subsequent users, other than resource utilization is almost free. Businesses should build services throughout the enterprise that can be reused within the company and potentially by its business partners. Reuse allows the project and engineering cost to be amortized over a greater footprint of users. This potentially increases the ROI of a service. In addition, single purpose applications that served one group of corporate users in the past may find it economically viable to offer services across the enterprise that exceeds the expected rate of return.
3. **Untangle expensive from cheap activities to lower process cost dramatically while increasing process speed.** When processes are not well understood it seems that all activities need to be completed by the same resources whether it is structural (e.g., computers, networks, software) or human capital. Low cost activities cannot be moved to cheaper locations or be done by lower cost IT equipment or labor. For example, do all the process activities need to be done on

High Availability hardware or do all X-Rays need to be reviewed by a local and perhaps expensive radiologist? During the SOA effort, the physical and information component may be identified as separable. This is important because high intensity information activities are more location independent and easier to outsource or distribute geographically than high physical presence activities. The key caveats to dispersed information related activities are both human and structural communication bottlenecks that could arise. Process discovery will uncover what needs to be done close in proximity to the business and what activities are independent of locality. SOA can drive the discovery process.

What are the Risks Associated with SOA- The Truth Comes Out?



Most of the risks in our view are related to people, not technology issues. The negotiation effort and skill amongst the LOB's management will likely determine the success of SOA. The next important predictor of success is the presence of managers and architects that can "broker" and execute the planning and delivery of the projects.

1. **Reuse talk is easy but takes intense discipline to execute.** Reuse of services is the corner stone of SOA but reuse has been "hocked" before as the key value of Object Oriented software – that effort largely failed. Object orientation is a concept that has been promoted since object oriented development became popular in the early 1990. Few enterprise users outside of software product companies received a good return on the approach, if any. This was largely due to the lack of maturity of software engineering at many enterprises. Tools were not inherently available to make it easy to discover components so they were very often rebuilt. The good news is the industry has learned from past deficiencies in standards and built into the SOA conceptual framework a discovery mechanism that can find services dynamically and easily.
2. **Who has been successful with SOA - big budgeted, very disciplined tech savvy management.** Companies with success in SOA to date have always had success with technology. Typically, they are big companies with big budgets and businesses that are technology intensive. These companies also have supportive and technology savvy managers. Many are in the financial services industry.
3. **SOA requires a significant centralized effort.** Without proper management at the enterprise level, SOA will lose traction. If your organizational structure is significantly centralized already then this is a mute point. To gain maximum benefit, management must look across the enterprise to understand what services are needed and how they need to be used. Generalization of those needs must be done so that all users are satisfied. Not an easy task. This is where centers with top architects and people with superb negotiation skills are critical.
4. **A LOB versus enterprise mindset stifles agreement on cost allocation to build services.** There is additional upfront cost to enterprise focused SOA services versus building traditional LOB type applications with no services capabilities. Those costs include additional management, increased negotiation efforts and the enhanced robustness required in offering services to a wider client base. However, there is a large potential for the organization to recapture that cost over the life of those services. The problem is most organizations incent managers so they are judged on performance at the LOB level not at the enterprise level. Also, they are typically rewarded for results in the near term - 12 months or less. This is likely driven by how investors value the share

price on the stock market, quarter by quarter. A major component of the SOA change cost will be how organizations measure their managers, much more as a co-operative team than individuals.

5. **Enterprises that rather “code it” than “design it” are prone to SOA failure.** Efficiency of scale through degree of reuse is highly dependent on design – something that few companies invest time or money in these days. Organizations in our experience have not enforced the execution of application design in the past and likely will have a difficult time instituting design in the future. Often, companies say they have a design phase. But with time and hiring budget constraints, few real engineers surface that can make/enforce a systematic approach for “**the hoard**”. This is especially true when it comes to abstract concepts. Systems and networks that are more concrete in nature than applications tend to have a significant design effort that is done well. Application development that should have a substantial design phase and use of abstractions often don’t. The software industry is populated by 80% concrete thinkers and 20% abstract thinkers. Unfortunately, there are not enough abstract thinkers to go around. As a result, for all but the best managed organizations, software engineering and SOA maturity are likely to be of too low of rigor to be effective.
6. **As with most IT projects, SOA initiatives are hard to quantify in terms of ROI with details down to the spreadsheet level.** Most managers are use to this fact. It is critical for those companies that have to mitigate many of the issues mentioned above because the cost of change and risk levels will be that much higher. That is, limited tech savvy management, few real Architects/Engineers, a highly decentralized organizational structure and poor LOB inter co-operation dynamics combine to conspire against the successful implementation of SOA.

What to “Hammer Out” first for successful SOA

The real question may not be **when but if**. Figuring out who is going to pay to get the services built and the other side of the cost, how is everyone going to be paid to keep the services running is critical – there are always priorities and it is usually differentiated by how much each user is paying.



1. **Services cannot be free - yet many companies are run like communist empires.** This may seem obvious that services cannot be free, but this runs counter to how many companies implement initiatives with IT. For example, we have seen Fortune 50 companies that implement a mission critical application such as a core insurance portal or an Internet banking application with little attention to prioritization of resource use. Often, the LOB applications access a data tier that allows any application to access resources and use as much as it wants – no priorities. What’s mind boggling is that a sales clerk running a massive database query can cripple customer’s throughput for online banking during peak hours. Why? Because there are no priorities for requests for services. Mind boggling #2 is that many staff don’t understand a basic tenet of human life – There has to be priorities because almost all resources are limited – your time, your money and your patience. Another way to think of it is from a formal business perspective. First, free services go against the progressive costing accounting approach of **activity based costing**. Second, no costing neglects organizational **priorities of allocating scarce resources** to the most profitable initiatives. The service owner’s applications can easily be squeezed out by “customers” from other LOB’s wanting services for “free”. This leads to...

2. **Who is going to pay for the services?** Ed the VP of Business Development wants services - If they are free, even better. Bill the VP of Marketing who offers services cannot just allow all his servers to be saturated by other LOB's requests. Bill wants to charge for them not only to recover real time costs but project, design and engineering cost associated with offering the service to other LOBs. Which leads to...
3. **If there is a charge to build it, then each LOB will want to get the data they want, their way.** So now, the requirements not only include the original interface for the service, but also it must incorporate all the requests of the LOB that plan to use the service. There may be a few or many requests initially. Companies need a mechanism to broadcast services to be offered like an RFP to get cooperation and feedback. There will likely be ongoing modifications required as the service sees more reuse. Once you get over this, the organization will need...
4. **More rigorous and centralized support in the form of portfolio management and architecture development.** For companies that are highly decentralized, especially on a global basis this may be a shock to the system. Then again, it might be a good excuse or driver for a CEO that wants to gain more control over a decentralized organizational structure. Which leads to...
5. **More overhead costs due to the centralized effort.** However, if the effort is successful, just like the overhead of creating a software product, this overhead cost can show worthwhile returns.

Technical Fundamentals - SOA and where the ESB fits in?



SOA involves the orchestration of repeatable business tasks or also known as services. The top technical drivers for SOA are standards, the pervasive ability to discover services and widespread interoperability between systems and applications. Although SOA is technology neutral, the best technology currently available to implement it is web services coupled with an ESB such as WebSphere Message Broker V6 and/or WebSphere ESB V6.

What is the SOA Lingo so I can sound like I know what I am talking about?

SOA from a business perspective includes:

1. **A service is a repeatable business task.** The key attribute is that services are published to a common location on the network. Any application wanting the service can inquire and receive information about that service. The published location is very much like a person that calls 411 or uses a telephone book to find a phone number. The key difference is the amount of information in the service directory (assistant) on the network. The directory has addresses as well as information about what can be requested in the form of a declaration outlining the available services and how to get them.
2. **Service Oriented Architecture** - An IT Architecture that supports service orientation. Web Services is an example of a service oriented architecture. However, SOA does not necessarily have to be implemented with Web Services but currently is believed to be the best way to do so.

What are the Top Technical Drivers behind SOA?

The top drivers of SOA are:

1. **Standards** – Broad adoption of standards streamlines communications between internal business units and out to business partners. Easier interoperability reduces time to engineer solutions, thereby, speeding time-to-market while reducing cost. Think how housing prices are driven down because we use standard window sizes, doorframes and plumbing components.
2. **Dynamic Discovery** – A big problem in both business and personal paradigms has been finding “stuff”. Where to find your tennis racket at home, where to get the best deal on your next stereo or finding information within an enterprise has always been a challenge for humans. Google has made a good crack at helping people find information on the web. Enterprise applications at most companies are largely hard wired when it comes to application information sharing – and hard to discover without getting a software developer to dig it out of the code. This means that if the location or “address” is changed, many applications have to be changed. This makes the enterprise inflexible.
3. **Interoperability** – Services are independent of any specific programming language, operating system or network.



What are the IBM Components that Typically Deliver SOA?

Figure 1 shows an architectural breakdown of services and products used to deliver SOA. Closest to the business is the delivery and coordination of services. The Business Service Bus offers Web Services to implement services called by the Business Process layer. At the lowest level is the Enterprise Service Bus, which acts as an information highway connecting service requestors to service providers.

IBM Service Oriented Architecture

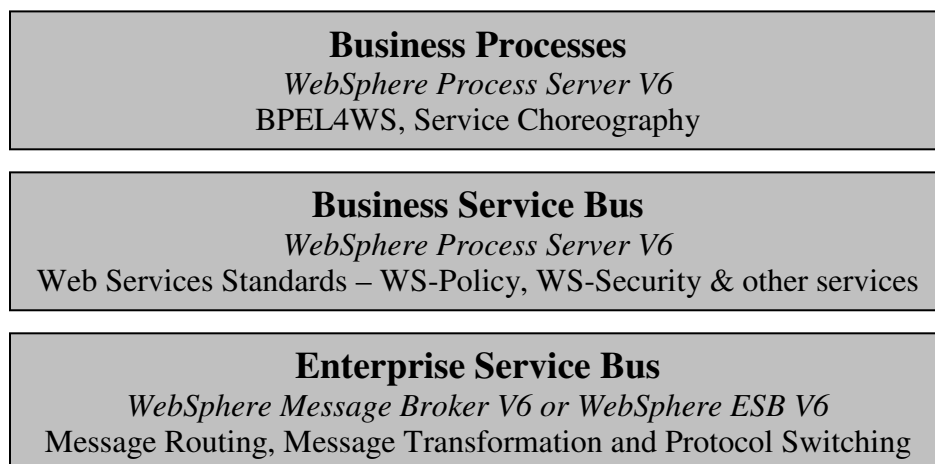


Figure 1 IBM Service Oriented Architecture

Contrarian View - Why its tough to be good - at Managing IT.



It is well known in cost accounting that there are four types of responsibility centers:

1. **Investment Center** – Latitude in rolling profits back into the company to improve performance. This type of center has the most amount of control.
2. **Profit Center** – No investment decisions, but profit decisions such as the best customers to serve are included.
3. **Revenue Center** – Responsible for revenue only – such as sales.
4. **Cost Center** – Responsible for costs only. The only hope of improving is through reducing costs. This type of center has the least amount of control.

However, many companies treat IT as a cost center. How can you tell? One sure “tell tale” sign is an accountant type is running IT. Hurry, go get the org chart. Accountants have a risk adverse psychographic profile. So innovating solutions that cannot be substantiated by NPV calculations is typically not acceptable (Net Present Value is the predominate method used by finance departments during project decision analysis). Another sign is that a CIO/VP has to report to a finance position such as a CFO. To be far, risk aversion is not always bad because this type of person or functional group can stop other groups or LOB from investing in projects that have little hope of achieving success - or worse, bankrupting the company.

Granted, not all corporations see IT as a cost center. Most of those that do, typically views IT as the cost of entry into a market, not as a means for competitive advantage or a point of difference to achieve premium pricing.

As a rule, people reduce stress through higher control in their jobs and personal life – this is almost instinctive. Therefore, what you will see is the best-trained managers gravitating to where they have the best chance to be successful and have more control over their future - that likely is in an investment center not a cost center. Salaries and bonus will be higher if they are successful – the upside is large for both good managers and the company. Conversely, in a cost center the only hope of affecting, at least directly, the bottom line is reducing cost.

So where do the best managers go if they are smart. Not IT, right!