Operationalizing SOA for the Composable Digital Business

David S. Linthicum,
Linthicum Research, LLC

Introduction

> ***A SOA is a strategic framework of technology that allows all interested systems, inside and outside of an organization, to expose and access well-defined services, and information bound to those services, that may be furthermore abstracted to process layers and composite applications for solution development. In essence, SOA brings the agility aspect to architecture, allowing us to deal with system changes using a configuration layer, rather than constantly having to redevelop these systems.***

The primary benefits of a SOA include:

1. Reuse of services/behaviors, or the ability to leverage application behavior from application to application, without a significant amount of re-coding or integration. In other words, SOA applications can use the same application functionality (behavior) over and over again, without having to port the code…leveraging remote application behavior as if it existed locally.
2. Agility, or the ability to change business processes on top of existing services and information flows, quickly, and as-needed, to support a changing business.
3. Most importantly, SOA allows the fundamental abstraction of application and infrastructure resources so they may be leveraged in the same way, at anytime, for any reason. This eliminates IT's need to deal with complex and ever-changing systems.

The motivation to move to Operationalizing SOA for the Composable Digital Business has many layers, including:

- Speed to Market
- Ability to Adapt
- Cost Efficiency
- Technological Advantage

Speed to Market applies when you leverage service-based resources. It's easy to allocate the resources you need to get a business moving in the right direction, and to move fast. This could apply to a new upstart, looking to get the right systems on-line to support new business processes, as well as to a large enterprise that wants to bring a new product to market.

SOAs value is all about the ability to provision the resources you need right now to run the business. We're moving away from the practices of waiting weeks or months for hardware and software to show up, bolted into the data center, configured, and then released into production.

SOAs have huge amounts of on-demand resources, sharable among those in the business who need to quickly move the business along. This translates into millions in additional revenue, as well as the ability to capture and lead markets. The latency around standing up computing resources has been largely removed.

Ability to Adapt is a bit different than speed to market, in that we're talking about the ability to change versus the speed of changing. For years, businesses have dealt with an IT infrastructure that's unable to change. For the most part, the culprit was the layer upon layer of technology. Technology was typically added to solve tactical problems, such as the need for an ERP system, a new database for business intelligence, or just to get the latest and greatest technology in-house. Integration into the overall system was often more of a patch than a fine mesh. The addition of each layer of technology caused more complexity. More complexity drove more cost, as well as the inability to change the IT infrastructure to adapt to changing business needs (see Figure 1).

Figure 1: IT has added layer upon layer of technology over the years, at the expense of agility.

The rise of complexity causes a few problems:

- First, costs go up. Complex infrastructures are much more difficult and costlier to maintain. More staff is required, more technology, and more resources to operate these systems longer term.

- Second, efficiency goes down. Overly complex IT infrastructures are much less efficient, and thus can't meet the needs of the business. These

suboptimal IT architectures require more resources for a diminishing return (see Figure 2).

- Finally, and to the point of this section, complexity resulted in IT infrastructure that lacks agility. The systems lack the ability to adapt to the changing needs of the business, and lack the latency required to change things that are complex.

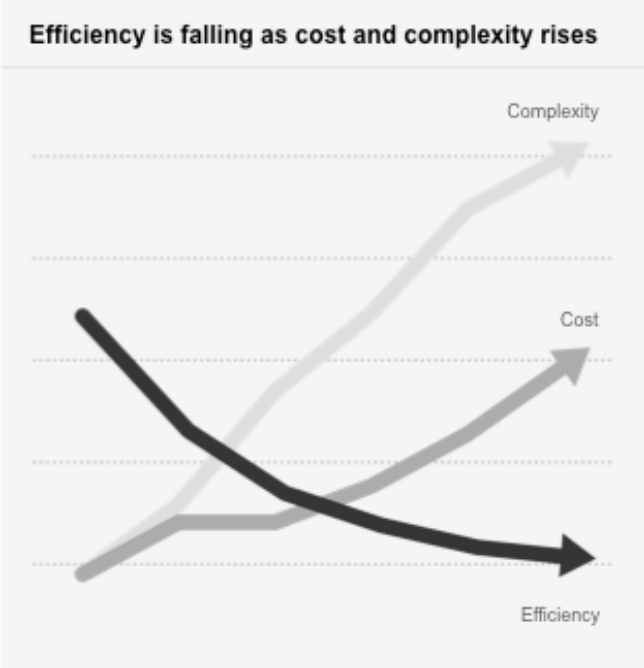

Figure 2: When IT infrastructure is too complex, efficiency falls as cost and complexity rises.

Business agility becomes the primary reason to move to SOA, at the end of the day. While providers and analysts will point to the ability of SOA to reduce costs, the ability to change the nature of the business is where enterprises typically find the value. This has been proven over and over again in the last several years, as we've been standing up the first SOA instances.

Cost Efficiency of SOA is the technology's ability to effectively utilize money. Again, this is really a secondary cost benefit of SOA, with agility (covered in the previous section) typically being the core benefit of moving to Operationalizing SOA for the Composable Digital Business.

There are a few areas to consider around cost efficiency in the context of SOA, including:

- ***The ability to operate at a lower cost of production***, meaning that we're able to operate applications, databases, and support end-users at a much

lower cost of production. SOA does a much better job of sharing resources. The more SOA shares resources, the lower production costs go.

- ***The ability to avoid risk*** is defined in much more detail below. However, for now, it's the ability to push risk onto SOA. Much like we push the risk onto our power companies and water companies, with SOA, we just pay for the services we leverage, inside or outside of the company.

- ***The ability to shift around technology changes***, which is a bit different than business agility. As technology changes over time, we have a much better platform to keep up with those changes, perhaps abstracting enterprise IT from dealing with those changes directly. Service producers are constantly upgrading software to provide better services, or expanded capacity. This occurs automatically.

As we can see in Figure 3, the idea is to reverse the issues we covered earlier. Move efficiency up by taking cost and complexity down. Operationalizing SOA for the Composable Digital Business holds the promise of providing the technology that can make this happen. However, while this is clearly the objective, the path to get there is rather complex. This is the purpose of this paper, to provide you the information needed to transform the way in which you approach IT within your enterprise.

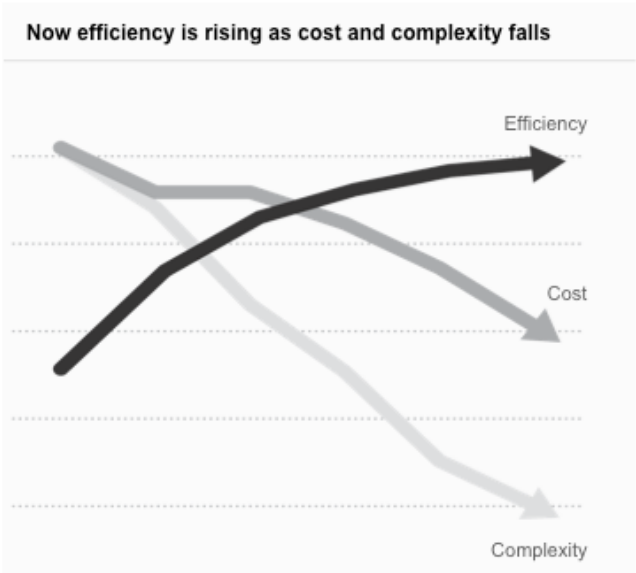

Figure 3: The objective of cloud computing is to increase efficiency and cost by lowering complexity, using a more effective way of delivering technology.

So, what are the attributes that we're seeking with Operationalizing SOA for the Composable Digital Business? It comes down to specific ***disruptive vectors***, or new emerging patterns that should be followed by enterprise IT. They are the

fundamental components of Operationalizing SOA for the Composable Digital Business, and should be considered as transformational methods that will fundamentally change the way that we approach the design, development, deployment, and the operations of enterprise solutions.

Disruptive Vectors

In order to set some lines in the sand as to what technology is useful, and what technology is now outdated, we've come up with 9 disruptive vectors. These vectors should provide sound 'how to' guidance to evolve your IT practices, as well as outright change the way we approach IT altogether.

Most of us have heard of these types of "revolutions" before. This is more about the evolving use of SOA, which provides for sound operationalization, and thus finally allows us to implement SOA in useful and productive ways.

Logical Views of Resources

This vector describes architectural patterns that allow us to take a logical view of physical, and non-physical resources. This means that we're able to place physical and virtual resources, using abstraction (covered next), in the order that best aligns with the meanings within our business. This could mean assigning semantics to logical views, or assigning specific labels.

The value of this vector is that it allows you to take views of IT resources, and place them into domains that make more logical sense to those who leverage these resources. This feature removes the application developer, designer, and architect from having to consider the origin of all resources; instead they deal with a common layer of semantics.

Layered Abstraction of Physical and Virtual Resources

To deal with virtual and physical resources, the use of an abstraction layer when building a SOA provides some key advantages. While it's almost always a good idea, an abstraction layer comes with different levels of value, depending upon the problem domain, the core business needs, and other factors.

The value proposition of an abstraction layer is its ability to loosely couple composite services, data, and applications from the underlying physical resources, including data and lower-level services, such as micro-services. However, physical and virtual resources can be anything that produces data and/or behavior.

This provides the value concept of agility, or the ability to change the physical or virtual resources without affecting processes, services, applications, etc.. As covered above,

agility is the core value proposition of SOA that allows architects to quickly align the architecture with the changing nature of the business.

Data-Driven Automation

Data-driven automation is a concept in which the processes describe the data to be matched and the processing required, rather than defining a sequence of steps to be taken. There are many benefits and problems with this paradigm.

Each function must know the abstract data type of the variables it is working with. Functions and interfaces can be used on all objects with the same data fields, for instance, the object's "position." Data can be grouped into objects or "entities," according to preference, with little to no consequence.

Virtually Centralized Governance

One can consider governance as the ability to define, track, and monitor service execution on any number of platforms, on-premises, or cloud computing-based. Virtual centralized governance allows us to further abstract ourselves away from the services. Because they are centrally controlled, governance at the resource as well as service level is much more logical and thus make us more productive.

The value of virtual centralized governance is clear, when you consider the amount of risk it removes. Those who manage the systems can be more proactive, and get well ahead of issues that will bring down services which, in turn, will bring down the systems.

Virtually Centralized Security/Policy

Centralized security/policy management provides a huge advantage when using complex and widely distributed systems. For instance, there are so many moving parts to SOA-based systems that security is a nightmare. The ability to assign centralized security and policies to all these parts — person, device, or data — allows for more control, as well as more flexibility.

Adding value to the concept of identity-based security is the notion of centralized identity management, or what I call centralized security and policy management. In short, this is the ability to provide credential validation services delivered from a central source. This is a mandate when creating and operating complex service-based systems, all of which need security to be systemic and centrally controlled.

Portability

Portability refers to the ability for applications, services, processes, and data to move from one platform to another. The degree of modifications that must occur

defines the degree of portability that the technology is able to offer. Clearly, the more portable the systems are, the more value they can bring.

When dealing with SOA, we certainly have the the ability to centrally leverage services from many platforms, and combine those solutions into applications. However, portability refers to the ability to move code and data inter-platform, such as from traditional systems to those hosted within cloud services.

Rapid Re-Configuration of Applications

Along with the advantages of leveraging SOA, we need to consider the efficiency around reconfiguration of applications. Applications that take too long for reconfiguration, slow down the ability to get back into production, and thus hinder agility and time to market (covered next).

For instance, in order to make changes to very long-running applications or those that must be continuously available, we must rapidly and dynamically change the application. Rapid reconfiguration of a distributed application is the act of changing the configuration of the application as it executes. This means dealing with state information, including transferring this information from the old configuration to the new configuration.

Compressed Time-to-Market

One aspect of agility is time to market, which SOA can compress because solutions are built from pre-built services. That means you can quickly deploy automated systems and quickly bring products or services to market. But shorter time to market is more valuable in certain industries than others; for example, it matters more in high tech than in manufacturing.

Resilience / Adaptability

Resiliency and adaptability is the ability for systems to recover from disasters, ranging from minor power outages to complete destruction of data center facilities.

Resiliency is often achieved through the use of redundant components, subsystems, systems or facilities. When one element fails or experiences a disruption, the redundant element takes over seamlessly and continues to support computing services to the user base.

Adaptability and resiliency go hand in hand, since systems that are resilient must also be adaptable. For instance, the ability for an application to reconfigure itself and operate on a redundant platform. Ideally, users of a resilient and adaptive system never know that a disruption even occurred.

Rankings

In order to provide better context, and evaluate the value of technology patterns, we placed the features and functions of EnterpriseWeb's technology in line with technology currently provided in the industry (see Table 1). The idea is to rank the EnterpriseWeb solution against the disruptive vectors we've identified above.

Disruption Vector		EnterpriseWeb	Industry
Logical Views of Resources		4.0	1.5
Layered Abstraction of Physical and Virtual Resources		3.5	1.0
Data-Driven Automation		4.0	2.5
Virtually Centralized Governance		3.5	1.0
Virtually Centralized Security/Policy		3.5	1.0
Portability		3.0	1.0
Rapid Re-Configuration of Applications		4.0	3.0
Compressed Time-to-Market		4.0	2.0
Resilience / Adaptability		4.0	1.0
Average		**3.7**	**1.6**

Table 1: Weighted scoring and vendor positioning between EnterpriseWeb and technology used in the industry to support SOA.

A few assumptions were made:

- Industry, mean an amalgam of most SOA solutions sold on the market these days, including those from larger providers such as IBM, Oracle, and Computer Associates.
- The rankings were objective, done by Linthicum Research, LLC, and consider only the features of the technology.
- The rankings were made by looking at various common use cases, including cloud systems, big data systems, IoT systems, etc..
- Technologies in this sector vary greatly, in terms of features, functions, and how they approach SOA.

Understanding the Rankings

As you can see in Figure 4 below, EnterpirseWeb technology exceled in each disruptive vector, with some very slight weaknesses in "Layered Abstraction of Physical and Virtual Resources" as well as Portability. The industry amalgam did not rank highly in any of the disruptive vectors, except it showed some strength in "Rapid Re-Configuration of Applications" as well as "Data-Driven Automation." However, rankings are well below the EnterpriseWeb technology in those categories.

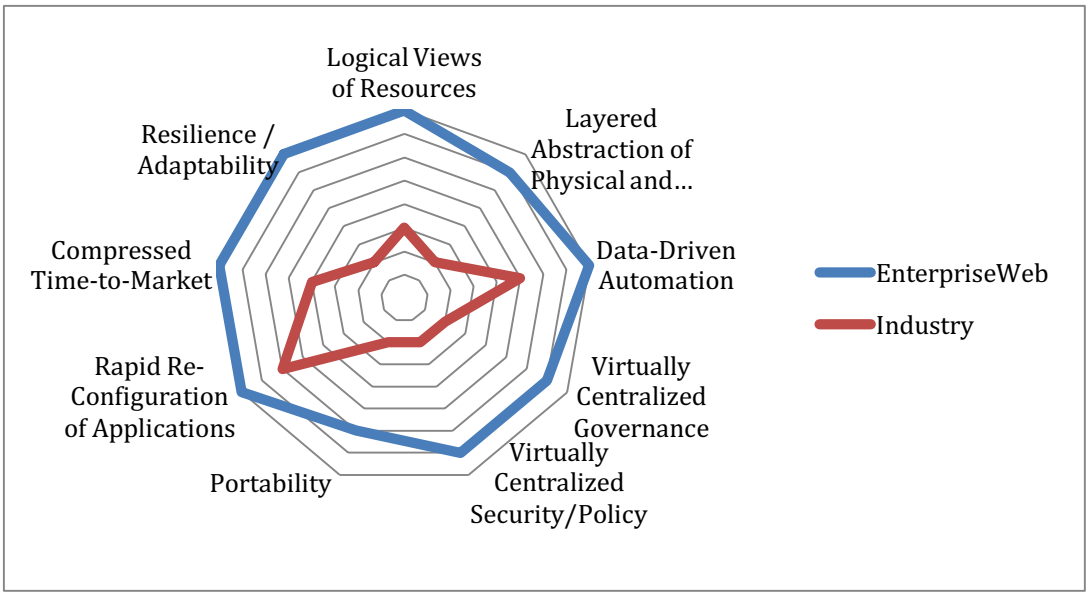

Figure 4: The enabling technology's ability to reach the outer limits of these disruptive vectors means an increase in IT capabilities.

Since all models in EnterpriseWeb are loosely-coupled, interprets them in real-time, it always fetches the latest objects and current state information for a response that is both dynamic and adaptive. Importantly, EnterpriseWeb also supports non-disruptive API evolution ("Dynamic APIs"). This is a key factor in many of the higher rankings, since it provides a better way to implement and operate services.

Dynamic APIs are models of interfaces, which: 1) de-couple business "intent" from technical implementation for event-based, data-driven, policy-controlled service realizations; 2) consistently communicate change semantics to enable APIs to evolve in dynamic environments as underlying partner services are updated, upgraded and replaced/substituted.

EnterpriseWeb's Dynamic APIs can be extended with non-functional concerns (e.g. security/identity, business compliance, IT governance, system controls, etc.) for rich transactions. Instead of inconsistent development processes that implement these

rules on a one-off basis, the policies are strictly enforced. These policies are explicitly stored as objects in the library. They can be referenced by application models as re-usable patterns, providing consistent, automated enforcement of organizational policies.

Steps to Change

What's core here is that we make stepwise improvements on how we do SOA architecture, which will improve how we build, deploy, and operate core systems. The concept we attempted to address in this paper is to not compare technologies, as much as get enterprise IT to think differently about how systems are designed, built, and maintained.

Therefore, we recommend the following steps:

First, determine your requirements and the business value of a SOA approach, and consider the disruptive vectors we've presented above. The business case should reflect savings in development, but, more important, the value that can be delivered throughout the years.

Second, make sure you validate your assumptions, and test all technology to determine its applicability in your own problem domains and use cases. Your SOA problem patterns are unique, thus your SOA solutions patterns should match up.

Finally, make sure to evaluate the success of your first SOA project. Create metrics to determine what should be measured, such as productivity, agility, cost, etc.. This data will be handy to help adjust the use of the technology and processes accordingly, to make sure your project is as productive as possible.

Call to Action

EnterpriseWeb supports orchestration of Cloud, IoT and Network Services, as well as dynamic human workflows, for end-to-end automation. It acts as a distributed computational fabric that provides unified management across silos, partners, technologies, protocols and domains.

The EnterpriseWeb platform enables dynamically connected value-chains that facilitate transformation to the composable, software-defined enterprise. EnterpriseWeb is the on-ramp for hybrid-Cloud. It simplifies distributed computing, while reducing the cost, effort, and risk of modernization.

Those who are moving toward SOA, or toward SOA running within the cloud, should make evaluation of EnterpriseWeb technology a core requirement. Consider the

findings of this paper, as well as your own productivity, when building and operationalizing SOA.

About the Author:

Leading technology publications frequently name David S. Linthicum among the top 10 enterprise technologists in the world. He is a true thought leader in the industry, and an expert in complex distributed systems, including cloud computing, data integration, service oriented architecture (SOA), and big data systems. As the author of over 13 books on computing with over 3,000 published articles, as well as radio and TV appearances as a computing expert, he is often quoted in major business and technology publications. In addition, David is a frequent keynote presenter at industry conferences, with over 500 presentations given in the last 20 years.