Service-Oriented Architecture: Myths, Realities, and a Maturity Model

Rudy Hirschheim
Louisiana State University (U.S.)

Richard Welke
Georgia State University (U.S.)

Andrew Schwarz
Louisiana State University (U.S.)

Executive Summary

Service-Oriented Architecture (SOA) is espoused as the next structural innovation within the IT marketplace. Our findings, based on interviews with fifteen key individuals responsible for SOA deployment at ten organizations, suggest that there is a ‘disconnect’ between the in-print prescriptions regarding SOA and what is actually happening. For example, the vast majority of organizations were using SOA as an IT architectural initiative rather than as a business transformation tool. Perhaps because of this technology focus, most organizations did not see SOA (and services in general) as a vehicle for solving larger business needs, but rather as ‘IT things to be consumed by IT.’ As this paper discusses, we do not believe that this is because SOA is merely ‘hype.’ Rather, most of the organizations participating in this study were relatively early in their SOA deployment strategy. We therefore propose a five-stage SOA Maturity Model with six corresponding dimensions (including both IT and business elements) and argue that higher levels of maturity will lead to “game shifts” as business and IT interests converge around SOA and are able to exploit its transformational aspects.

INTRODUCTION

The interest in Service-Oriented Architecture (SOA) as a “guiding principle” derives from an interest within the IT community to move from the large-scale development of applications to the creation of services that more directly map to business needs. The resulting shift from applications to functionality built from services, could ultimately alter the way IT relates to, and integrates with, business units within and across organizations. That is, an SOA approach will result in IT working more closely with the business units and partners, not only on service definition and development, but also on the associated business process redesign necessary to deliver the functionality desired by business units.2

As organizations define, develop and deploy services, the resulting services will be available for end users to amalgamate (or mashup) these services into end-user, situational applications in ways that the developers may not originally envision. Creative use has already been evidenced by early adopters of SOA; websites such as Google, Ebay, and Amazon are exposing services for end users to utilize and, combined with other services, to create novel applications. For example, users on the web have created applications that allow the public to map real estate listings using Google Maps, track weather using webcams and forecasts, and other innovative applications (see http://www.programmableweb.com/popular for a list of popular mash-ups). Enterprise mash-ups such as those built using Denodo’s platform enables the agile creation of new services by integrating existing data from a variety of sources, not only structured and internal data, but also unstructured or semi-structured external content (www.denodo.com). In short, SOA places customizable power in

1 Robert Zmud and Carol Brown are the accepting Senior Editors for this article. This report is based on research sponsored by the Advanced Practices Council of the Society for Information Management. We gratefully acknowledge their support.
the hands of the users and better facilitates active participation by end users in the IT development process.

Consider the following example of HousingMaps.com. Currently, Google exposes a number of its applications as services. HousingMaps.com (www.housingmaps.com) is an example of a composite service which uses Google’s web services and Craigslist’s (www.craigslist.org) information about available houses. HousingMaps.com merges data from Craigslist on houses for rent, sale, and sublet with Google Maps yielding a spatial view of where the available houses are located. An interested buyer or renter can click the bubbles on the generated map and get the address, price, photos, and other details about the listing. The information can also be sorted by different cities and price ranges. Before SOA, this service would have required developers from HousingMaps.com to have close ties with both Google and Craigslist to program, integrate, and manage data between these applications. One can easily imagine the coordination of technologies, versions, and processes that would have been necessary to both develop and maintain the composite application; an amazing feat. Because Google and Craigslist expose their functionality in the form of web services (the primary building block of SOA), it is possible to have a loose coupling of two discoverable services allowing the creation of an entirely new and valuable service. With SOA, HousingMaps.com can integrate these services without having an explicit relationship with either Google or Craigslist. As this example shows, SOA offers opportunities which are new and exciting with the potential for significant business advantage. And it was for this reason, that the current research project was initiated. The project fundamentally sought to answer the question: Are companies moving to SOA and, if so, how and with what effect?

The paper is organized as follows. First, we explore what SOA is and its architectural features. We consider why firms would choose to adopt SOA. Next, we outline the research project—who was involved in the project and what research questions drove the project. Then we report on the findings that emerged from our data collection and analysis. We note some surprising results which conflict with what the so-called experts are saying and consider why this might have been the case. Finally, we conclude that there exists an SOA maturity curve and most of the firms we interviewed were still positioned relatively early on this curve. This suggests that SOA could still be the transformational tool that the proponents contend, but its transformational aspects only begin to surface after a certain level of maturity has been achieved.

WHAT IS SOA?

Service-Oriented Architecture is an IT architecture where data and logic functionality are “black boxed” or encapsulated with only their input and output exposed for others to use. What best differentiates SOA from its predecessors—such as function calls, application program interfaces (API’s), objects, and components—is a set of open standards that defines what this encapsulation (service) is, how to communicate with it, and how to find and use it, as well as the ability, in principle, to both perform and consume these services anywhere the Internet exists.

As with SOA’s predecessors, there are a number of reasons for placing data and application functionality into service containers: (i) the ability to re-use them; (ii) the flexibility to reconfigure, improve and re-assign them without breaking the overall delivered functionality; and (iii) the ability to provide transparency across multiple (legacy) applications and data sources at a lower unit operating cost. Any chunk of functionality can be circumscribed, black-boxed and defined as a service. An architecture that specifies design principles for laying out and developing the collection of services is needed. As organizations that deploy SOA are finding, these guiding principles for service definition should be based on business needs for these services. However, most organizations are not presently structured as a pre-defined set of services.

SOA, then, is becoming more than just an IT architecture with services being defined by IT in response to ongoing operational needs and business functional requirements. SOA is driving the need for a broader business architecture from which higher-level (coarse-grained or composite) services can be more readily identified and used to guide the definition and planned development or purchased availability of lower-level services. A few organizations have begun to adopt the view of becoming a Service-Oriented Enterprise (SOE). One such organization is Intel, whose architectural vision is provided below (Figure 1).[

---

3 We thank Matt Levy (Research Assistant at LSU) for pointing out this example.

4 This figure is adapted from the presentation used by George Brown, IT Research, Intel Corporation, to discuss the company’s SOA strategy, Jan. 2007.
ARCHITECTURAL COMPONENTS OF SOA

According to Dornan, SOA is fundamentally about middleware; exposing server functionality as standardized services that can then be reused in new applications. It requires some hardware or software to mediate between the services and the application that is accessing them. These intermediaries are typically divided into a number of product categories such as ESBs (Enterprise Service Bus), governance products (involving a combination of a catalog and a source code management system), SOA management systems (embodiing a network management framework), and security gateways.

At its core, SOA is a particular approach to the overall design of how computer-based functionality is packaged and delivered to consumers of this functionality – other computer applications, business processes that need access to specific functionality, business users as portal or mashup building blocks, and external touch points (e.g., vendors, clients, service providers) seeking to connect directly with a company’s internal data and processes. Its distinguishing characteristics are:

1. All functionality to be consumed by various clients is fully encapsulated as a set of services with a defined description, method of invocation, method of communication, and well-specified descriptions for the data messages sent and received.
2. Service declarations, descriptions and invocations adhere to a set of published, open standards to allow for interoperability, within and across organizations.
3. A service registry, where definitions and connection points reside to allow standardized service discovery and transparent re-direction.
4. Service-related specifications and their invocations are expressed using XML.

How these basic principles are realized will differ from organization to organization as there are many SOA-related standards from which to choose. To provide a consistent and interoperable basis for building SOA-conforming capabilities, an overall

source: Intel
standard called WS-I (Web Services Interoperability) exists to specify some basic choices. For example, it requires that:

- The service be described according to a particular standard (WSDL or Web Services Description Language)
- The invocation of the service be done using a specific message type (SOAP or Simple Object Access Protocol) that in turn communicates with a SOAP processor,
- The method of communication or “transport” be HTTP (and hence the term “web service”)
- A particular protocol be used to discover services and their invocation address (UDDI or Universal Description and Discovery Interface).

There are, however, competing service definition and execution standards (e.g., Representational State Transfer—REST). When all of these are realized in a set of software, sometimes called the “service stack,” this collection is termed an Enterprise Service Bus or ESB. To conform with what has become a de facto standard, we’ll use the term “web service” from this point forward to distinguish more general (business) services from those implemented according to the standards of HTTP-based SOA. Beyond this basic collection of functionality and standards prescribed by WS-I, there are many layers that can and often do exist above the ESB or its equivalent. For example, an “orchestration” layer is a piece of software that’s capable of wiring together and executing a collection of web services in order to achieve a more aggregate (called “composite”) functionality. Here too, various standards exist for the wiring language, such as BPEL (Business Process Execution Language). The processor that uses BPEL is then often referred to as the BPEL or process engine. Its purpose is to orchestrate a set of services, according to the BPEL specification provided, to achieve a higher-order functional result. For example, using a set of internal database services, external credit-check services and other rules services, to achieve the resulting composite service: Verify_Customer_Order. Wiring together pre-existing web-services in this way is commonly termed “integration-centric” orchestration.

Other standards exist to extend the basic (SOAP) service definition into areas such as security and reliability. As more experience is gained with web services in increasingly complex situations, additional open standards have been, and continue to be, proposed to address these issues – underscoring the value of an open standards approach. As well, other styles and types of orchestration languages have been developed that involve both human and computer-based actors – so called “human-centric service orchestrations,” along with process engines that understand and can execute these specifications. Examples of such specifications include XPDL 2.1 and BPEL 2.0.

Another dimension is “meta-data” (data about data) and its management. This arises from the need to operate across many areas of the organization with many underlying data sources and varying transaction-types, each giving the same referent name to dissimilar data or, conversely, the same data-type being named differently. To reconcile these differences and consistently perform the necessary transformations on the underlying (XML) data, a set of meta-data specifications and their associated transformations is needed.

Business rules are another area that can be extracted from orchestrations and other forms of business process specifications to provide the ability to more easily change them in synchronization with organizational desire or necessity. By pulling out all of the higher-level conditional logic into their own rules set and keeping them in a common rules store, they can be changed as needed, on the fly, without having to touch the underlying composite service definition or the equivalent execution logic. Generally there is a separate business rules processor and management system that effects this by providing specialized rules services that are then consumed by an over-arching service orchestration of the types described above.

A rapidly emerging layer—associated with the storage, management and use of events that are a readily available by-product of service execution—is referred to as an “event bus” and its architectural component as Event-Driven Architecture (EDA). Events can be used as a mechanism for triggering specific services (event or exception handlers), as part of an overall process orchestration, or as real-time insight into what is or isn’t happening within an organization. Complex Event Processing (CEP) languages and processors now exist to provide a single way in which to define and use these events for everything from alerting and business activity monitoring (BAM), through real-time business intelligence (BI) and onto what is called “autonomic computing” or the ability of processes to sense and respond (self-adapt) to changing situations—the true definition of agility.
To complete this excursion “up the stack,” there is the human interface layer where web services are directly exposed to users. These take the form of portals (with the web services being used as “portlets”), mash-ups (as previously discussed) and various forms of business intelligence displays (such as dashboards). Associated with each of these higher-level layers there are adopted, proposed or emerging open standards.

It is impossible to draw an overall architectural stack that would be widely agreed upon because of the diversity of implementation approaches to basic web service delivery, the adoption of additional layers of functionality that build on the existence and use of web services, and the way in which vendors and organizations choose to blend existing software services, such as applications servers, message-oriented middleware, and transaction marshalling and queuing. Nevertheless, in Figure 2 we attempt to graphically summarize an SOA architectural stack.

**OUR RESEARCH STUDY**

Based upon the apparent widespread interest in SOA, a research project was initiated by the authors to determine what is really happening within organizations deploying SOA. Our expectation was that we would find a diverse range of experiences with SOA uptake and deployment, given the diversity of standards, standards adoption, SOA architectural elements and layers, pre-existing applications and infrastructure, motivations for adopting SOA, and stages of adoption. And with this diversity, one might equally expect a variety of insights and opinions regarding SOA and its current applicability. We also wanted to investigate the degree to which SOA has become the basis for a new, renewed or extended dialog between IT and the business, given the presumed need at some point to align the web services development roadmap with business process, service, data view and business intelligence/alerting needs.

The stated objective of the research study was to analyze the drivers, processes, and challenges, and how these challenges can be overcome. The research approach was to conduct interviews with 10 organizations that had had varying degrees of experience with SOA implementations but all of whom were considered “advanced” in their use of information technology (i.e., acknowledged sector
leaders). Fifteen individuals were interviewed, with each interview lasting between 45 and 90 minutes. In many firms, the enterprise architect was the key individual we interviewed; but in a number of organizations we also interviewed CIO’s and CTO’s. We included a wide range of industries, including pharmaceuticals, food, insurance, manufacturing, government, energy, healthcare, airline, retail, and banking. As expected, the firms did report a range of SOA experiences – from companies just beginning to some with six years of SOA experience. However, we did not detect any specific industry differences.

FINDINGS ON SEVEN RESEARCH QUESTIONS

While there are any number of reasons for the espoused wisdom on why and how firms pursue a SOA strategy, the answers we received from the interviewees ran somewhat counter to published “wisdom.” In general, we found that there is a disconnect between ‘espoused’ views by industry researchers and consultants, and ‘actual’ practice, based on the responses from our practitioner interviewees. In the following, we will therefore juxtapose our practitioner findings with the SOA espoused views. Where appropriate, we include interviewee quotes to highlight these findings.

Our findings are discussed in terms of seven questions we developed to examine the why, how and what of SOA adoption in organizations. (The complete interview guide can be obtained from any of the authors.) The seven questions addressed are:

1. How are services / SOA defined?
2. What are the initiating reasons that motivate companies to adopt SOA and what role do business stakeholders play (if any)?
3. How is the SOA initiative justified?
4. What are the specific methodologies and approaches used to successfully deploy SOA?
5. How is SOA sourced (with respect to individual services and enterprise platform)?
6. What changes to IT governance are required due to the SOA deployment?
7. What are the future directions organizations are likely to take in terms of individual services and ESB as they build upon the collection of SOA services deployed and the additional functionality this affords them?

1. How are services / SOA defined?

Espoused View. The espoused view is that a service is the encapsulation of specific transactional business functionality with a well-defined interface “contract” that is specified using an open standard (e.g., WSDL). A service is simply a software component that can be accessed via the Internet and provides clearly specified functions. A few years ago we would have called a service a ‘software component,’ but perhaps a more accurate view would be to think of a service as a software component designed to be used in an Internet environment.

SOA, on the other hand, is both architecture and supporting software. The architectural service component is aligned with business architecture, while the software suite supports standards-based web service discovery, encapsulation, invocation, and messaging. According to Foster and Tuecke, SOA is a set of information systems architecture principles that enable the creation of IT functionality by combining loosely linked and interoperable services (e.g., add name and address, purchase item), which can be reused in different combinations and presented to users in different ways to reflect how they want to work. Enterprise architects believe that SOA can help businesses respond more quickly and cost-effectively to changing market conditions.

What We Found. We found a disconnect between the espoused view and those of our interviewees. Generally, the interviewees considered any software component wrapped in an industry-defined interface (WSDL, DCOM, ASP.Net, etc.) to be a service … “Services are nothing new; we’ve been doing services with DCOM, CORBA, etc. for over seven years.” Further, we found that while “architecture” for many interviewees (but not all) was “key,” the type of architecture varied (software, process, information, service). We further found that formal SOA software (e.g., a full enterprise service bus or ESB) was not widely adopted: one-third are using it; one-third are considering/evaluating; and the remaining third will consider it sometime in the future. Many chose to emulate parts of a more formally defined ESB: “We realize we will need a service registry at some point, but as yet we haven’t chosen one; we use Excel as our current service registry.”

---

2. What are the initiating reasons for SOA?

**Espoused View.** The espoused view is that there are five reasons why organizations initiate SOA projects. First, SOA enables increased business agility. By logically decomposing applications into a collection of services (“servitizing” the application), the services can be re-combined and reused in a manner that allows the firm to more flexibly respond to changing functional needs. Second, SOA bridges the gap between business processes and IT solutions. By decomposing applications and associated data access into a collection of services, these services can more readily support the processes within the firm, leading towards a tighter link between business and IT. Third, SOA facilitates greater reuse. IT developers, rather than building from scratch, can reuse previously defined services to meet new or changed requirements. In turn, reuse brings lower cost, a shorter implementation schedule, and reduced risk. Fourth, SOA facilitates the incremental insertion of new technologies because they can be inserted as needed, leading to plug and play versus rip and replace. Finally, SOA increases operational flexibility.

**What We Found.** Our research discovered that agility and flexibility were not the primary drivers of SOA projects although, in some cases, it was a secondary effect derived from the other motivations. We also found that there was little relationship between services and business process redesign. Of the five espoused initiating reasons identified above, none were primary drivers of the interviewees.

We found that the SOA projects among our sample of companies were initiated for a different set of five reasons:

1. Infrastructure manageability: “We do in the vicinity of 2,000 EDI transactions and we are using it [SOA] to integrate the EDI workflow with the different applications that we have.”
2. Standardization of resources: “We share a lot between divisions ... and we are trying to get more formal about services and sharing services and SLAs.”
3. Creation of a formal method for re-use.
4. Interaction with trading partners.
5. Master data management within the enterprise: “SOA means a core commitment to standardizing our data assets across the enterprise and standardizing our processes across the enterprise because to achieve the reusability and standardization and agility we are looking for, you first have to tackle standardizing your data and standardizing your processes.”

3. How is the SOA initiative justified?

**Espoused View.** The espoused view is that SOA is a strategic initiative that involves both business and IT. Some of the benefits derived from this point-of-view were enumerated above and summarized here: (1) flexibility and agility; (2) inter-organizational, value-stream improvement; (3) a common view on key entity information; and (4) improved business processes and customer touch-points.

The expectation, then, is that the justification for SOA infrastructure deployment and training should be a separately funded organizational investment, rather than being boot-strapped on a project-by-project basis. Development costs for services rest upon getting the business to understand the need for improved services in support of new or improved business processes and customer-facing applications.

**What We Found.** We discovered that companies are not getting the business units involved in the SOA initiative. Some IT respondents were even more adamant about keeping SOA totally transparent to the business side. In organizations where SOA initiatives are underway, we found that IT is either absorbing the costs internally using a set-aside fund or rolling SOA infrastructure and development costs into IT projects because SOA cost justification is difficult. Example statements regarding how SOA is funded include: “We absorb the costs of SOA as part of other projects’ budgets” or “We let our ERP drive our SOA initiative.”

4. What are the methodologies and approaches used?

**Espoused View.** New service definition methodologies, e.g., IBM’s SOMA and extensions to UML 2.0, are seen as the basis for service definition and construction related to a software-driven approach to service creation (so-called bottom-up approach). Various business process development and improvement methodologies are used to define the primary clients for the to-be developed services leading to service definition driven by business process needs (so-called middle-out approach). For a few organizations, service thinking has become a

---

way of viewing how the organization itself should be configured, giving rise to what is called the Service-Oriented Enterprise (SOE). Often this comes about as a realization that a roadmap is needed to guide the definition and creation of services, rather than adopting a bottom-up (technology need) or middle-out (business process need) approach to defining services. Adopting a top-down view on service definition requires a different set of methods such as IBM’s Component Based Development (CBD) or Venice Consulting’s RIVA to define and prioritize the high-level service compositions the organization will need.

What We Found. We discovered that most of the services created so far were fine-grained services used within and among software applications (e.g., Oracle’s facilities to ‘wrap’ DB calls) and thus adopting a bottom-up approach to services definition by default. Service definition and implementation was primarily achieved using existing software development methods (“We don’t have a formal (SOA) method – we use Rational and VS.net”), with some considering the use of IBM’s SOMA methodology. The decision about what services to implement varies by organization. For most, the services roadmap was driven by software integration and maintenance needs with very few indicating they had over-arching processes, or even higher-level SOE architectures, to guide their current service development efforts. There was, however, a recognition by some of the need for this: “[Defining services] is next to impossible to do project-by-project; [you] need to evangelize solutions across projects and processes.”

5. How is SOA sourced?

Espoused View. The sourcing question has two dimensions. First, how are organizations sourcing their SOA platform – the connectors, message brokers, registries and other components that comprise a functioning service delivery infrastructure? Second, how are organizations sourcing the individual (web) services themselves? The espoused view is that companies should choose best-of-breed standards and services including web services and associated infrastructure software.

At the infrastructure level, one can choose among a number of full SOA platform vendors, each with its own ESB offering (e.g., IBM’s WebSphere family of SOA servers, Oracle’s Fusion, TIBCO’s ActiveMatrix, Progress Software’s Sonic ESB, Software AG’s CentraSite, etc.). Organizations are expected to evaluate, and then select the best-of-breed vendor(s). At the individual web service level, services could be insourced or multi-sourced depending on corporate strategy (i.e., core versus non-core), with a recognition that service creation increases IT expenditure in the short term, but produces long-term cost savings.

What We Found. With respect to the infrastructure level, most interviewees did not consider an ESB platform a key part of SOA, nor did they have one in place; only a few were in the throes of adopting one. Rather, they used bits and pieces of their existing infrastructure in combination with some specialized SOA functionality. We also found that the selection of an SOA platform appeared to be influenced more by the particular vendor with which the company was dealing (e.g., its ERP vendor) than with business processes or organizational strategy. The most commonly reported solution to dealing with increased SOA infrastructure costs was to build the cost into existing projects.

With respect to the sourcing of web services, several organizations indicated that they are outsourcing the development of web services to third-parties … “We will likely give this task over to some outsourcer once we define our services.” Few acknowledged the use of third-party provided services (i.e., web services that run on third-party servers and are accessed over a WAN or the Internet).

6. What changes to IT governance are needed for an SOA deployment?

Espoused View. The espoused view is that services need to be centrally controlled through a centralized governance architecture.8 Beyond centralization, the espoused view is that SOA requires its own set of policies. The definition, monitoring and enforcement of these policies are increasingly viewed as critical to long-term SOA success. New policy and control mechanisms are, for example, needed in areas such as security, reliability and integrity, ownership, and version control over the service life-cycle. The espoused view generally is to recommend some type of Center of Excellence for service/process development and governance to act as a centralized clearing house and point of control.

What We Found. We found that SOA did lead to an interest in SOA governance, but none of the companies shifted completely to a centralized governance architecture and few had developed

---

specific SOA governance policies … “We tried to standardize the processes for SOA in terms of our overall methodology for executing enterprise architecture across projects.” Some put new centralized governance structures in place (e.g., an SOA Council) to handle policies. Others recognized the need for new policies, but there was no clear consensus on how to achieve this. SOA implementations did not result in new roles being created. We found that the SOA implementation was largely driven by an architecture group when such groups existed, resulting in increased responsibilities for this department. Finally, we found two side effects for SOA projects: they led to standardization of project management processes and to a questioning of corporate governance: “When we start to talk about sharing services and we take it up to the level of business services, then you are going to see some conversations going on from our enterprise IT governance group around the business governance process.”

7. What are the future directions that result from SOA?

Espoused View. The espoused view is that SOA, coupled with BPM, can become the primary architecture for unifying existing applications and delivering new business functionality. The thinking is that currently rigid packaged and homegrown ERP, SCM, CRM and HRM systems are transformed into flexible business processes by re-constituting them as a set of business services wired together by process orchestration. This notion is further supported by ERP-type vendors who provide a collection of web services (e.g., SAP’s NetObjects and Galaxy) that tap into their application software’s underlying functionality and can be used as building blocks for organization-specific composite web services and business processes.

Beyond business processes, other commonly cited business benefits of SOA are the prospect for a unified informational view of business entities, leading to real-time business intelligence and activity monitoring. Web services can be employed at the line-of-business level to create situational functionality delivered in the form of web service mash-ups. In sum, the espoused view is that SOA provides a basis for a new, common dialog between IS and the business related to services.

What We Found. We found that the primary business interest expressed was more a unified view of data across multiple applications (“Create a full-view of [business entity] using SOA”). None of the companies we interviewed had an immediate plan for BPM, BAM, or events, although some indicated future interest in these aspects of SOA use—expressed in such statements as: “[Our] Long-term view is to model business processes and associate with services.”

None of the firms we interviewed had used SOA, services, BPM or BAM as a basis for creating a new dialog with management over the use of IT, but we did find an acknowledgement of a need for this dialog: “SOA is being driven by IT but we’re much more aware of business needs than before.” Those who had explained SOA to their business colleagues almost uniformly used mashups as their exemplar for “why SOA,” although none indicated immediate plans to develop or use web services for creating mashups.

PROPOSED SOA MATURITY MODEL

We suspect that the differences in practices between leading organizations and the espoused wisdom regarding SOA can be explained by a company’s stage in SOA maturity adoption and use.9 We also suspect that much of the public rhetoric on SOA is not entirely self-serving or based solely on hype.10 Rather, these espoused views of SOA are likely to be informed by certain best practices that those proclaiming them have observed during their consulting and operational experience with SOA. In many cases consultants and professional researchers act as intermediaries between best practice and/or first-mover companies and everyone else. Thus, the gap we found could be attributed to where the companies in our sample were in terms of SOA maturity relative to other companies that have managed to achieve higher levels of SOA maturity.11

In the practitioner literature, one can find a number of SOA success stories (Bank of Canada, Credit Suisse, }

9 We acknowledge that we are not the first to propose a model of SOA maturity. See, for example, Arsanjani, A. and Holley, K. “Increase Flexibility with the Service Integration Maturity Model (SIMM): Maturity, Adoption, and Transformation to SOA,” IBM Developerworks, Sept. 30, 2005. The SIMM approach suggests seven levels of maturity focused solely on the technical aspects of each stage. In contrast, our approach examines both the IT and business dimensions that correspond to each stage.

10 A sentiment shared by others, such as Krill, P. “Industry Report: SOA is Overly Hyped”, InfoWorld, Aug. 20, 2007; and, more recently, Linthicum, D. “Taming the SOA Hype Monster,” InfoWorld, June 18, 2009.

eBay, Amazon, Citigroup, Deutsche Post, Wells Fargo Bank, and Winterthur, among others) that suggest that there is much more potential to SOA, especially at the business level, than was found among the organizations surveyed in our research to date. To be sure, the firms that participated in our study all had sophisticated and experienced IT units and, in some cases, were acknowledged industry leaders, but that doesn’t appear to be either a necessary or sufficient condition for discovering the comparatively few who have moved further along the SOA maturity cycle.

Based upon our study, we hypothesize that there are five stages of SOA maturity through which organizations progress in their SOA initiatives. These are graphically depicted in Figure 3 above. In the initial stage of SOA maturity, organizations use fine grained software components. As the organization matures to the managed stage, standardization of data and resources emerge and standardized software architecture begins to emerge. As the initiatives continue, the firm moves in to the defined stage and begins supporting business processes with services, including process redesign. Next, they move into the quantitatively managed stage where the focus is on achieving key objective measures, which allows the enterprise to create a service architecture designed to promote flexibility and agility. Finally, during the optimized stage, business processes are created that lead to an adaptive architecture.

Using this maturity lens, we propose six dimensions of SOA: view of SOA; SOA benefits and metrics; involvement of business managers; SOA methodology; where services are sourced; and governance. Based upon where an organization is in its SOA maturity, the firm has a corresponding set of attributes for these six dimensions, as shown in Figure 4.

Using this maturity model as a framework, we placed the 10 companies we studied on the maturity model grid based on the responses to our interview questions. As Figure 5 suggests, the majority of the firms were in the first two stages for each dimension. Only a few of those interviewed were in the Defined stage and none were in the two most mature stages.


13 We acknowledge that the labels of our stages are congruent with other models of maturity (e.g., SIMM).
Our position is that more “mature” (and successful) SOA-oriented companies: (i) have the support and understanding of business; (ii) are defining services in business rather than IT terms and have both governance and architecture units in place that reflect this; and (iii) have value drivers that are not primarily efficiency-related (e.g., reuse) but instead (or also) reflect business value (e.g., common data view, support for business process management and business intelligence, etc.). A critical issue revolves around how SOA is viewed by the organization. Is the perspective on services—and the general approach to SOA deployment—technical or business driven? In other words, are services ‘IT things to be consumed by IT’ or ‘business things to be enabled by IT’? We expect that at some point along a hypothetical maturity cycle, business and IT interests converge around SOA and what it can support, and the “game shifts.”
CONCLUSION

Service-Oriented Architecture is widely reported as the next IT-based structural innovation that is being—or will be—adopted by organizations at some level. Our research determined that there is a substantial gap that currently exists between the espoused view and those of adopting companies. It is tempting to interpret this gap as the latest example of vendor hype—dismissing SOA as yet another seemingly minor change in how applications are built by IT. We believe such an interpretation is shortsighted.

We suspect instead that many organizations must go through an IT-driven (earlier) stage before evolving to a later stage with a greater business orientation, where the real business benefits of SOA appear to lie. Some may even choose to remain at these early stages in the absence of a clear and apparent reason or roadmap for moving forward. If our SOA maturity model in Figure 3 is representative, firms may need to get to stages 4 or 5 before they can capture the hoped-for business value on which the espoused wisdom seems to be anchored.

---