Break on Through to the Other Side: A Missing Link in Redefining the Enterprise

By John Hagel III and John Seely Brown

Over the past year, just as the initial hype over e-commerce has subsided, a new chorus of promises about the potential of the Internet has been gaining volume. The singers this time are not dot-coms and their promoters, but rather the big providers of computer hardware, software, and services. What they're promoting, through a flurry of advertisements, white papers, and sales pitches, is a whole new approach to corporate information systems. The approach goes by many different names—Microsoft calls it ".Net," Oracle refers to "network services," IBM touts "web services," Sun talks about an "open network environment"—but at its core is the assumption that companies will soon buy their information technologies as services provided over the Internet rather than own and maintain their own hardware and software.

These services, which we will call "web services", have some intriguing attributes. Unlike existing web sites designed for people to interact with information, web services connect applications directly with other applications. They do this through a form of "loose coupling" which allows connections to be established across applications without customization. As a result, these connections can be established without regard to technology platform or programming language and thus make it easy to share technology resources flexibly across a broad range of users and companies.

No doubt, many executives are skeptical. They've heard the big promises and the indecipherable buzzwords before, and they've wasted a lot of time and money on Internet initiatives that went nowhere. This time, though, there's an important difference. The technology providers are not making empty promises; they're backing up their words with massive investments to create the infrastructure within the enterprise needed to make the new IT approach work. As the build out continues, a steady stream of new web services will come on-line, providing significant cost-savings over traditional, internal systems and offering new opportunities for collaboration among companies. Slowly but surely, all your old assumptions about IT management will be overturned.

In this paper, we will provide a CEO's guide to the new IT strategy. We will explain what the new web services architecture is, how it differs from the

traditional IT architecture, and why it will create substantial benefits. We will also lay out a measured, practical approach for adopting the new architecture – a step-by-step approach that will pay for itself while mitigating the potential for organizational disruption. Indeed, we believe that two of the great advantages of the web services architecture are its openness and its modularity. Companies don't need to take a high-risk, big bang approach to its implementation. They can focus initially on opportunities that will provide immediate efficiency gains, incorporating new capabilities as the infrastructure becomes more robust and stable.

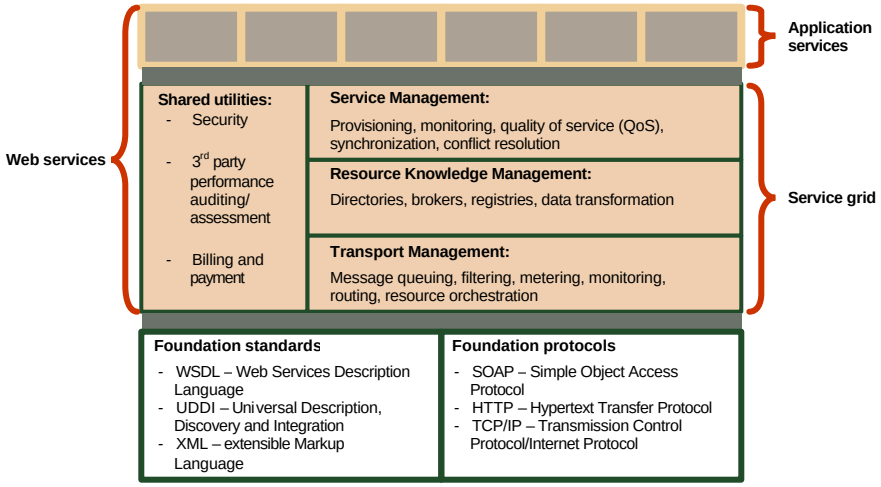
The New Architecture

Up until now, companies have viewed their information systems as proprietary. They buy or lease their own hardware, write or license their own applications, and hire big staffs to keep everything up and running. This approach has worked, but it has not worked well. After years of piecemeal technology purchases, companies inevitably end up with a mishmash of disparate systems spread throughout their different units. Over the last decade, in an effort to merge these "data silos," many big companies have invested large amounts of money—hundreds of millions of dollars, in some cases—in massively complex enterprise resource planning systems, which offer suites of interlinked applications that draw on unified databases. The ERP systems have certainly solved some problems, but they've created others. Because they're relatively inflexible, they tend to lock companies into rigid business processes. It becomes hard, if not impossible, to adapt quickly to changes in the marketplace, and strategic restructurings, through acquisitions, divestitures, and partnerships, become fiendishly difficult to pull off. In effect, the companies that have installed ERP systems have replaced their fragmented unit silos with more integrated but nonetheless restrictive enterprise silos. Even this understates the challenge. In fact, few companies have fully standardized on a common ERP system across all their units. Instead, they still grapple with a hodge podge of hundreds, if not thousands, of incompatible systems.

The web services architecture is completely different. Constructed on the Internet, it is an open rather than a proprietary architecture. Instead of building and maintaining their own internal systems, companies rent the functionality they need –whether it's data storage, processing power, or specific applications – from a network of outside service providers. Without getting too technical, the web services architecture can be thought of as comprising different layers of technology.

The foundation of this architecture starts with a set of communication protocols that help to establish connections across applications wherever they reside.

WEB SERVICES ARCHITECTURE



Adopting the Internet's basic transport protocols (TCP/IP and HTTP), the foundation adds a messaging protocol – the Simple Object Access Protocol (SOAP). SOAP acts as an envelope for simple, formatted messages that are automatically generated to call and invoke services. Using SOAP, programmers do not need to custom design messages to reflect the unique characteristics of each application or service being addressed. The only requirement is that each application involved has implemented a SOAP interface.

The second part of this foundation involves a set of standards that provide a common format for establishing shared meaning. The most critical standard is XML (eXtensible Markup Language). XML is a pre-requisite to let a service or application "read" the contents of a message. For example, XML will provide a "tag" indicating where a message refers to price. Once again, the objective is to deploy a common format that all applications or services can adopt so that programmers do not have to customize connections to the vagaries of individual applications. Now programmers can implement one "translation" from their application to XML, rather than having to implement a separate translation for every application that must be connected. XML can be extended to establish formats for a variety of other connection needs. For example, XML is being used to define standards for describing how to connect to specific web services (WSDL) and establishing a registry of web services (UDDI).

These protocols and standards create a foundation for a second layer of the web services architecture – the service grid. This service grid is analogous to the electrical power grid in that it provides a set of utilities and services to ensure reliable distribution of web services. Shared utilities within the service grid provide security, third party auditing and performance assessment and billing and payment services. The service grid also includes a set of managed services that will help to facilitate transport of messages, locating and understanding characteristics of available web services, and ensuring web services perform as required. The basic concept is to delegate to specialized service grid utilities two key roles (1) provide the demanding performance required to connect web

services together to support mission critical business activities and (2) help web services users and providers find and connect more easily with each other. Once again, the focus is on reducing the complexity for programmers seeking to connect web services. The service grid eliminates the need to reproduce basic functionality that all connections require and makes these available as specialized, shared services. This allows connections to be established more quickly and at lower cost, but without sacrificing on performance.

The service grid and foundation layers of the architecture support a growing array of application services. These might be application services that reside entirely within one enterprise and are shared across, for example, multiple business units. They might be application services developed by one company in the course of operating its own business and offered to other enterprises on a subscription basis as might be the case, for example, with a bank that develops a credit card processing application and offers it to other banks. Finally, specialized providers might develop these application services and as their primary business offer them on a subscription basis to other companies.

To illustrate how the architecture works, let's contrast the way a typical business activity – say, loan processing by a bank– would be carried out through a web services architecture versus a traditional proprietary architecture. Loan processing involves six discrete steps: capturing data regarding the applicant; validating the applicant data (i.e., verifying employment and bank/investment balances); credit scoring; risk analysis and pricing; underwriting; and closing. Loan processing typically involves interactions with other institutions, for example, checking with other financial institutions regarding applicant bank/investment balances and arranging underwriting of loans by other financial institutions. In a traditional IT architecture, the loan processing activity is usually supported by a single application operated within the bank. Interaction with other institutions is either handled electronically over leased communication lines with substantial proprietary technology required at either end of the communication line (as in EDI or ACH networks) or handled by people who manually enter the data at either end of the transaction.

A web services architecture changes this approach. First, the service grid will make it easier to connect at lower cost with a broader range of institutions. Rather than leasing dedicated lines, companies will be able to rely on specialized service utilities to provide comparable reliability, security and performance over the Internet. Also, instead of requiring each end of the communication line to install expensive, proprietary technology to facilitate communication, a web services architecture relies on public protocols and standards that are lower cost to implement. As a result, a bank will be able to connect electronically with a broader range of institutions and reduce the need to rely on human interaction and manual data entry.

This leads to a second difference. Rather than relying on a single application to perform the entire loan processing activity, the bank will be able to source major modules of the application from different providers. Since the web services architecture defines public standards and protocols for communication across applications, any application providers adopting these standards and protocols can interface quickly and easily with each other. So, a bank could decide to rely on a different third party provider for its risk analysis and pricing module and more easily integrate it with the other modules of the loan processing application. It might even decide to integrate multiple providers of this module since one provider may do a better job of risk analysis for loans to restaurants while another provider handles risk analysis for health care institutions better. As a result, banks would now have more flexibility to choose "best of breed" providers for each module within the broader application rather than compromising on performance because of the complexity of trying to integrate proprietary applications in traditional architectures.

A third difference is that lower cost and comparable performance of connections across institutions reduce the need to install these modules within the bank. The bank can choose to access these modules as a service delivered directly from the application provider's facilities. Specialized resource orchestration services will be able to help the bank develop and implement the rules to govern when a particular loan application needs to be routed to one risk analysis service provider versus another.

A fourth difference is that the bank can adapt more readily to innovation. Let's say it decides to expand its loan processing activity in the construction industry. Rather than modifying a large, traditional loan processing application at great expense over long time periods, it can search for providers of the modules that are particularly affected by such a shift (perhaps risk analysis and pricing as well as underwriting) and quickly and cost-effectively integrate these web services into its existing application portfolio using public standards and protocols. Specialized resource knowledge management utilities in the service grid such as directories, brokers and registries can help the bank identify and negotiate the terms of interaction with these new web services.

The web services architecture provides important advantages over its predecessor. First, it represents a much more efficient way to manage information technology. By allowing companies to purchase only the functionality they need when they need it, the new architecture can substantially reduce the investments they need to make in IT assets. And by shifting responsibility for maintaining systems to outside providers, it reduces the need for hiring IT specialists, which has become a significant challenge for many companies. Web services also reduce the risk of obsolescence; it will be a competitive necessity for third-party utilities to offer the most up-to-date

technology. Companies will no longer find themselves stuck with outdated or mediocre applications and hardware. The standardized, plug-and-play nature of the architecture will also make it much easier for companies to outsource activities and processes whenever it makes economic sense.

Second, and perhaps even more important, the web services architecture supports more flexible collaboration, both among a company's own units and between the company and its business partners. When traditional information systems needed to talk to each other, they did so through dedicated, point-to-point connections. For example, when a sales force management application needed to send information to a human resource application (such as closed sales for each salesperson to calculate commissions), a programmer had to develop a "connector" tailored to the proprietary interfaces of the specific applications at each end of the connection. Often, changes to the applications at either end would require further modifications to the connector software. The problem with point-to-point connections is that they are fixed and inflexible and, as they proliferate, they become a nightmare to manage. . Imagine a multinational business meeting attended by business people from 5 different countries, each one speaking a different language and not understanding any other. As many as 10 translators might be required to get any business done.

With a web services architecture, the tight couplings of the past are replaced with loose couplings. Because everyone will share the same data standards and connection protocols, applications will be able to talk with other applications much more freely, without the need for costly reprogramming. For example, in a loosely coupled connection, the information is translated into XML standard formats and transmitted using SOAP protocols so that the ability to connect is no longer dependent on understanding the proprietary interfaces of each application. In the same way that English in many parts of the world has become the lingua franca of business, these web services protocols and standards reduce the need to translate across each pair of applications by expecting everyone to adopt a common set of protocols and standards. As in the example of English, it becomes much easier to accommodate new business participants as long as they can communicate in English. This loose coupling makes it much easier for companies to shift their operations and partnerships in response to market or competitive stimuli.

Up until now, most companies that have done business on the Internet have had to yoke together existing systems with new ones to create the illusion of integration. A visitor to a corporate web site may think he's seeing a streamlined, integrated system, but behind the scenes there are often people manually taking information from one application and entering it into another. Such "swivel chair" networks, as they have come to be known, are inefficient, slow, and error-prone. Merrill Lynch, for example, had to struggle to patch

together 2,000 different systems to support its sites for clients. John McKinley, the company's chief technology officer, draws an analogy to the Potemkin villages in Czarist Russia, where brightly painted facades hid the unseemly reality of run-down homes. The web services architecture solves this problem. Taking the people out of the network, it will enable connections between applications – both within and across enterprises - to be managed automatically.

Swivel chair integration is necessary because of the significant expense and lead-times required to develop programs to integrate applications. Remember our analogy to the multinational business meeting? Imagine the expense involved to install dedicated connectors for each pair of systems when you are dealing with 2,000 different systems. For pairs of systems that interact frequently and in stable ways, the expense and lead-time of the dedicated connector may have been justified in the past. For systems with less frequent or stable interactions, inefficient swivel chair integration becomes the only viable option within traditional architectures. Recent approaches to Enterprise Application Integration (EAI) enhance traditional architectures by implementing specialized software to simplify connections across applications within the enterprise, but even these approaches encounter difficulty when the application integration task extends beyond the enterprise.

By defining public standards and protocols for integration, a web services architecture requires each application to implement these standards and protocols as part of its interface. Once this is done, the cost and lead-time for connection with another application adopting the same standards and protocols is much lower. The initial investment to implement these standards and protocols is amortized as more connections are established, in contrast to a traditional architecture approach where each new connection across applications involves a new investment. Given the much lower cost of integration, it now becomes feasible to automate connections across applications that only interact infrequently. Under the web services architecture approach, the existing applications continue to operate as before (at least, until it becomes more attractive to replace the applications with services offered by specialized providers) – all that changes is the implementation of a new, standardized interface to make connections with other applications easier

First Steps

The construction of the web services architecture is still in its early stages. It will take years of investment and refinement, as well as a good deal of trial-and-error, before it achieves its mature, stable state. That does not mean, however, that companies should wait to begin the transition to a new IT strategy. Even today, there are benefits to be gained by moving to a web-services model for certain

activities and processes. But it does mean that companies should take a pragmatic, measured approach. Fortunately, the web services architecture is ideally suited to such an approach. Because it's based on open standards and leverages the capabilities of third parties, companies don't have to place big bets at the outset. They can carefully stage their investments, learning along the way.

Merrill Lynch's McKinley, for example, is leading a number of initiatives designed to take advantage of web services. Two of Merrill's initiatives to deploy web services relate to the development of a new portfolio analysis platform to be used by Merrill's brokers and certain customers of its investment services.

The first initiative is to use XML to link together disparate systems residing both within Merrill and within key business partners. By defining standard XML formats for presentation of data, Merrill will be able to implement a much more flexible and lower cost way to integrate customer profile data, Merrill Lynch's product data and real time market data. This new platform will allow a secure, real time, integrated view of all the elements required to meet a customer's needs at any given time.

The second initiative employs a web services approach to leverage existing applications and present information from them on a variety of access devices, including computers, mobile devices and conventional telephones. By using XML to define standard presentation formats, Merrill will significantly reduce the cost and effort required to enable its brokers and customers to access information more conveniently, at any time and any place.

Merrill's experience, like that of other early adopters such as General Motors and Dell, offers three useful guidelines for other companies looking to get a head start:

Build on your existing systems. The web services architecture can initially be viewed as an adjunct to your existing systems. Through a process we call node enablement, you can use web or application servers to connect your traditional applications, one at a time, to the outside service grid. Node enablement can be as simple as just creating an explicit record of the interface of an existing application (known as an application programming interface, or API), along with its name, its Internet location and the procedures for interacting with it.. Node enablement leaves existing applications intact but "exposes" them so that they can be found and accessed by other applications in the web services architecture. This process of node enablement should proceed in a systematic manner, driven by near-term needs and opportunities but shaped by a view of longer-term business opportunities.

General Motors provides a useful example of this process. Mark Hogan, the president of eGM, a business unit created by General Motors to oversee its

consumer Internet initiatives, is a strong advocate of the web services architecture. Like Merrill Lynch, he began with fairly conventional web sites connecting GM with customers and dealers. He has used the ability to move incrementally into the web services architecture to target business areas with significant economic payoff and staged his initiatives to balance impact with effort required. For example, GM has developed a road map for generating more revenue and leveraging assets through its "Build to Order" initiative GM started by enhancing its traditional "Build to Stock" business processes with functionality like "Locate to Order", where dealers can more effectively respond to customer orders to locate GM cars already in dealer inventory. This first step of their phased approach reduced delivery lead-times for GM by 40% in North America. Further stages involve implementing "Order to Delivery", designed to shorten lead-times from the time a customer orders a car to the time it is actually delivered, and ultimately moving to "Build to Order", where customers can more easily order customized vehicles and receive them quickly and reliably.

The ultimate payoff is enormous – the long-term goal is to cut in half GM's $25 billion inventory and working capital investment. Goldman Sachs estimates that web services based supply chain initiatives can reduce GM's operating cost per vehicle by more than $1,000. Yet this staged approach allows GM to shift its IT architecture incrementally, focusing only on the systems required to deliver economic payback at each stage of the deployment.

The web services architecture plays a central role in implementing this sequence of business initiatives. Mark Hogan points to the fact that GM needs to collaborate more effectively with more than 8,000 dealers in North America, representing a broad range of technology platforms and many possessing very limited IT skills in their own operations. In this environment, the ability of dealers to connect more effectively with key GM systems without substantial investment of their own is critical. "If we can't offer a substantial and rapid business payback to our dealers, they won't adopt the platform." Equally important is GM's ability to increase functionality rapidly in frequent increments. Hogan insists that "conventional IT architectures simply aren't up to the task -- web services architectures provide the only way to rapidly enhance IT platforms" Since the applications will be delivered as web services, they can be easily and cost effectively upgraded at the central web server rather than having to send out upgrade software to thousands of dealers and worrying that some dealers might still be operating on older versions of the software.

Hogan also values the flexibility offered by web services architectures in areas such as GM's innovative OnStar service offering to over one million subscribers. "GM couldn't have predicted accurately the range of services OnStar might eventually encompass – a web services architecture allows us to rapidly innovate and build upon our early lead in this area." For example, if GM discovers that

subscribers value shopping information services more than financial information services, a web services architecture makes it easier to recruit and integrate additional shopping information service providers. Since interfaces are based on XML and SOAP standards, the GM and the new service providers will not have to develop proprietary point to point connectors, but instead can integrate new services into the OnStar offering much more quickly and at lower cost . The speed and low cost of integration make it easier to recruit new service providers and also enhance the ability to experiment with new services. If proprietary point to point connectors were required both GM and new service providers would think long and hard before deciding to add a new service to OnStar.

Dan McNicholl, the CIO for GM North America, recognizes the power to move incrementally with a web services architecture as well. He outlines a phased migration approach but he understands that key elements of the web services architecture, including such basic building blocks as XML standards and segmented approaches to security, are still evolving. He is managing the migration through incremental stages geared to the evolution of the architecture itself. McNicholl is clear on the end result: "it is not a question of whether we move to this new architecture, but when."

Start at the edge. In designing an incremental approach to implementation of a web services architecture, early adopters are concentrating their initial efforts on applications and business activities at the "edge" of the enterprise. By edge of the enterprise, we mean areas that collaborate most closely with other enterprises or consumers. Certainly sales and customer support would be examples of edge activities. But so is procurement and supply chain management that seeks to more effectively integrate the activities of suppliers. In some cases, what may previously have been completely internal to the enterprise may move to the "edge" and require collaboration with other enterprises as a result of outsourcing initiatives. For example, in the electronic industry, an increasing amount of manufacturing activity is being outsourced to specialized manufacturing service firms, creating a need to integrate manufacturing applications across enterprises.

Why is there so much focus on the edge? One key reason is that it is precisely at the edge of the enterprise that the limitations of existing enterprise IT architectures become most apparent. Applications in these areas are most likely to benefit by being shared across multiple enterprises and firewalls and, as a result, they are often the first to experience challenges that current IT architectures are not well-equipped to address – for example, connecting across heterogeneous technology platforms where there is no single management control point.

Let's go back to GM's North American dealer network. The 8,000 dealers in this network represent a significant challenge for any technology architecture. At least 15 different systems are used across this network and some of the smaller

dealers don't use any computer system at all. Many of GM's dealers handle cars from multiple manufacturers and operate multiple locations. Under a traditional technology architecture approach, GM would have three choices: impose a single system on all its dealers, develop customized connectors for each system represented in the dealer population or resort to swivel chair integration, relying on lots of people at high cost to integrate operations across incompatible systems. Each of these options imposes high costs (although they might differ in terms of who bears the cost) and are relatively inflexible once implemented. The appeal of the web services architecture in this case is two-fold: it allows GM to deliver sophisticated application services to its dealers without requiring them to install and maintain additional applications on their premises and, by using public standards and protocols, it creates low cost connections that enhance flexibility both for GM and its dealers. Take the example of GM's dealers that represent multiple manufacturers. Rather than having to adopt expensive proprietary technology required by each manufacturer, dealers can create one interface for their applications using web services standards and protocols and integrate these applications with any manufacturer adopting the same public standards and protocols.

Within the enterprise, the CIO has an opportunity to impose a certain set of standards governing the use of IT platforms. As we move into inter-enterprise applications like supply chain management or sales channel management, many enterprises must participate, each with its own technology platform. Of course, if one business participant has substantial market power, it can impose a certain technology platform on its partners (as has been the case in the implementation of EDI), but this approach has shortcomings. For example, it only really works if the business participant with market power is first to impose technology platforms on its partners. Most business partners must themselves deal with a large number of other enterprises. As they find themselves drowning in the complexity of complying with conflicting technology mandates of multiple enterprises, they are likely to become increasingly resistant to demands for yet another platform.

It would be far more pragmatic to adopt a technology architecture that accepts heterogeneous technology platforms within each enterprise and facilitates connection across multiple platforms. This approach becomes particularly compelling as the number of business partners increases and the nature of their relationships rapidly evolves.

These challenges at the edge are worth overcoming. Inter-enterprise applications offer the potential to deliver significant business value in terms of operating cost savings, asset leverage and accelerated innovation cycles. To do this, they must integrate with the business processes of other enterprises. High performance

companies increasingly see the opportunity to move beyond the enterprise in search of the next wave of efficiency and growth.

In this regard, Dell provides a great example of "edge" initiatives in its efforts to collaborate more effectively with direct material suppliers. Direct material purchases represent as much as 70% of Dell's revenue, so even modest savings in the area can deliver significant bottom line impact. Equally important to Dell is the inventory asset exposure – in an environment where product prices recently have been declining at 0.6% per week, any excess inventory can become very costly. It is not surprising that Dell sees significant benefits to more effective supply chain management.

Dell began by focusing on its network of third party logistics providers who operate distribution centers ("vendor managed hubs") where raw material inventories are maintained for Dell's assembly operations. Dell's "ship to" target from the time of receipt of a customer's order is 5 days, yet the average fulfillment time of their suppliers is 45 days. To meet their "ship to" targets despite the 45 day delivery time from suppliers, Dell used three business approaches at the outset: maintain a 26-30 hour inventory buffer at the Dell assembly plant, ensure vendors maintain a 10 day inventory buffer at the vendor managed hubs and distribute on a weekly basis a 52 week demand forecast to suppliers. The vendor managed inventory at the hub helps to deal with unanticipated supply disruptions and inaccuracies in the near-term forecast.

In the first stage of its initiative, Dell focused on reducing the 26-30 hours of inventory held at their assembly plants. Using i2 supply chain software, Dell now generates a new manufacturing schedule for each of its plants every two hours to reflect customer orders received over the previous two hours. Publishing those manufacturing schedules as a web service via Dell's extranet, Dell alerts the vendor managed hubs regarding what specific materials are needed and directs them to deliver those materials to a specific building and dock door so that the materials can be fed into a specific assembly line. These alerts are sent in an XML format so that they can be integrated directly into the disparate inventory management systems maintained by its vendor managed hubs. The distribution centers are implementing the capability to respond to the material requests automatically through the extranet and then have 90 minutes to pick, pack and ship the required parts to the factory. With this automated, web services-based approach, Dell has been able to reduce its own holding of raw material inventories from 26-30 hours of production to 3-5 hours – a reduction of more than 80%. Eric Michlowitz, Director of Supply Chain E-Business Solutions for Dell explains, "what we've been able to do is remove the stock rooms from the assembly plant, because we are only pulling in materials that we need specifically tied to customer orders. This enabled us to add in additional production lines, increasing our factory utilization by one-third."

Of course, lean manufacturing approaches often push back inventory exposure from the manufacturer to the supplier. Dell's goal is to eliminate excess inventory, not simply push it back up the supply chain. This drove its next wave of initiatives designed to eliminate the need to hold buffer inventories in the vendor managed hubs because of errors in the supply chain. To do this, they focused on checking the reliability of supplier delivery schedules early enough in the process to allow effective contingency planning to mitigate against unanticipated disruptions in supply (e.g., by temporarily withdrawing certain models on Dell's web site). Michlowitz's team is deploying a web services-based event management system to do this through its extranet. This approach avoids the need to directly access the application systems of its suppliers and instead relies on an approach consisting of automatic exchanges of inquiries and confirmations with its suppliers (e.g., inquiring whether the supplier shipped as promised). Because the system is automated, the supply chain team is able to focus its energy on handling the exceptions. Dell expects to be able to reduce inventory exposure in the vendor managed hubs by 10 – 40%, while at the same time significantly improving gross margin performance through more effective matching of demand and supply.

Michlowitz points out that the benefits of more effective integration across the supply chain extend beyond operating cost reduction and asset leverage. "New product introductions represent one of the most potentially disruptive events in a supply chain. In a fast moving industry like ours, Dell must manage 2,000 product transitions each year. Supply chain event management becomes essential to handle this pace of innovation." Since new product introductions quickly render inventories of components for older products obsolete, one of the biggest challenges in managing new product introductions is to reduce the risk of inventory write-offs, both for Dell and its suppliers. By sharply diminishing the inventory of components required at the third party logistics suppliers and at Dell's assembly plants, supply chain event management reduces inventory exposure during new product introductions. By focusing on collaborating more closely with its suppliers and logistics providers through supply chain event management, Dell found that it had to integrate more effectively into its own assembly operations. Dell's rapid growth and geographic expansion spawned a network of assembly plants running on a diverse set of legacy applications. To integrate supply chain operations through a web services architecture, Dell used the XML framework to define a common name for each part, module and sub-assembly used in its plants, along with a common way of describing the key attributes of each product component. Once Dell created this common language within the XML framework, then machine-readable electronic documents associated with the supply chain (e.g., a bill of lading) could be exchanged between Dell and its supply chain partners. By enabling the creation of documents that could be read by applications, Dell is helping to automate the processes that span across multiple enterprises while still preserving the

applications and databases already installed. Thus, initiatives that began at the edge of the enterprise to integrate more effectively with suppliers inevitably expanded to include a broader range of activities within the enterprise. While early implementations of web services architectures tend to concentrate at the edge of the enterprise, they have the potential to penetrate and redefine enterprise IT architectures more broadly as managers begin to understand the value of integrating a broader range of business activities using this approach. As a result of its XML initiatives, Dell is now better able to coordinate activities across its assembly plants since it now has a common reference language to describe its products and managers can more effectively compare product status across a very diverse set of plants and legacy applications.

Michlowitz is a strong proponent of the loosely coupled approach that is central to the web services architecture. He maintains that relations with business partners must rapidly evolve and believes that loosely coupled approaches are essential to support 90-day implementation cycles. "The loosely coupled approach also minimizes the technology burden on business partners and increases not only the capability, but also the incentive, for collaboration." Business partners have an incentive to collaborate with Dell because the company is not requiring that the business partners adopt expensive technology platforms as a prerequisite for collaboration (as in the case of traditional EDI networks)or to develop proprietary connectors that can only be used to connect to Dell. Instead, business partners at most need make only modest investments in node-enabling their applications, an investment that can be justified not only in terms of collaborating with Dell, but also with any other business partner adopting the web services architecture as a basis of collaboration.

Build shared meaning All of the early adopters wrestle with one of the key challenges involved in implementing a web services architecture. An enterprise CIO can impose within the enterprise at least some small degree of shared meaning, by requiring standardized terms (e.g., accounts will always be referred to as "ACCTS") and establishing a common set of definitions governing the meaning of these terms Constructing such shared meaning across enterprises is a much more organic process that cannot be imposed from the top down.Incremental implementation of this web services architecture, starting with a few long-standing business partners and expanding outward over time to embrace a broader range of participants, becomes critical to building this shared meaning. John McKinley at Merrill Lynch argues that the ability to build shared meaning becomes a key governor of the speed of adoption of this new technology architecture. This is one of the main reasons that broad third-party market makers have foundered – what at first seemed like a simple transaction turned on subtle distinctions. Master distributors in markets understood the

shades of meaning among different buyers and sellers and were able to use this knowledge to more effectively broker transactions. For example, a produce distributor knows the different ways each of its suppliers qualify the ripeness of an orange and, in turn, also knows the different ways each of its customers evaluate the ripeness of an orange. By creating a standard rating system and promoting its adoption within the community of suppliers and customers, the produce distributor can help to build shared meaning across the community.

It also explains why early efforts to implement web services architectures have tended to concentrate on private trading exchanges. Within close networks of business partners like GM's dealer network or Dell's third party logistics providers, it is much easier to build on existing trust and shared meaning and expand out over time. In effect, XML standards are giving rise to the formation of communities of businesses focused on defining more specific meaning within particular business contexts.

XML does this by providing a framework that can be extended into ever more detailed levels of meaning. Earlier in the paper, we used the analogy of English as a lingua franca for business to describe the role of standards like XML. Actually, this analogy is not exact. XML establishes a common grammar, but it leaves the need to specify semantics (i.e., which words have which meanings?). It is certainly helpful to know that a certain XML tag refers to the price of a product, but is the price a net price after discounts? Does it include shipping? The exact meaning of price needs to be defined and broadly adopted.

XML acts as kind of a scaffolding for creating shared meaning across a particular business community by providing the formats for specifying key terms along with their meaning. Within this XML framework, community members make proposals and argue out a set of common definitions that are shared across that community. As experience is gained in the use of these definitions, shortcomings will be discovered, leading to the continual refinement of these definitions

One of the roles of the service grid is to provide a set of utilities focused on building shared meaning within specific business communities. In many cases, dominant companies within private trading networks will provide these utilities. In other cases, industry consortia like the ebXML Initiative and RosettaNet will take the lead in developing XML formats for specific markets. RosettaNet for example is focused on defining and promoting the adoption of standard XML formats to describe business process elements for the supply chain of the electronics industry so that, for example, all participants use the same terms to describe activities like issuing a purchase order. These utilities might also be provided by independent businesses that are solely focused on developing XML formats.

A natural increasing returns dynamic will drive the evolution of the web services architecture. In the early days, incentives for adoption are limited because relatively few resources can be accessed through this architecture and the functionality of the service grid is still evolving. In this period, early movers like Merrill Lynch, GM and Dell play a key role by enlisting their business partners with a compelling business case. Over time, as more resources become accessible, the benefits of adopting the architecture become even more compelling. This will help to expand shared meaning, as later participants find it to their advantage to adopt meanings that are already in use in order to enhance access to a broader range of resources.

A Platform for Growth

Early implementations of the web services architecture have tended to focus on delivering additional efficiency by more effectively integrating business partners. This is completely understandable, particularly in an economic environment characterized by weak demand and growing margin pressure. Businesses will need to exploit aggressively the efficiency benefits of this architecture. In particular, leading edge adopters of this technology architecture are likely to be aggressive, high performance businesses that have already reengineered enterprise level business processes to achieve significant operating efficiencies. They will seek the next wave of efficiency benefits from two sources: more effective integration of business processes across enterprises and increased opportunity to outsource business activities and entire business processes.

These efficiency-driven initiatives are necessary, but hardly sufficient. The real beneficiaries of this new technology will be the enterprises that harness its power to drive accelerated growth strategies. Growth opportunities will arise in multiple forms.

At the outset, enterprises should aggressively explore opportunities created by this technology architecture to expose core competencies and make them available, for a fee, to others. Most managers, when first exploring the capabilities of the web services architecture, tend to quickly fall into the "consumer" mindset – in other words, focusing on how their enterprise might benefit by accessing resources from specialized providers of web services or business partners. What they overlook is the opportunity to generate revenue by taking activities where they are world-class and using this technology architecture to build new businesses by offering these activities as services for others.

Citibank is an example of a company moving quickly to exploit these revenue-generating opportunities. Citibank realized that electronic market exchanges need to rapidly add services beyond simply connecting buyers and sellers and executing transactions. In particular, in most market exchanges, participants must put up with the inconvenience of executing a commerce transaction (i.e., agreeing on the product, price and delivery terms) on one platform provided by the market exchange, but then handling the payment either offline or through specialized banking networks. Citibank has a well-developed skill in electronic payment services and saw an opportunity to increase convenience for buyers and sellers while at the same time expanding its reach as an electronic payment services provider.To address this opportunity, Citibank introduced CitiConnect, its electronic payment processing service using a web services architecture to automate connections across multiple existing application platforms. Machine-readable XML messages provide the "glue" that connects the functionality of the market exchange platform with the CitiConnect application and Citibank's Global Settlement Network, consisting of a variety of specialized Automated Clearing House (ACH) networks.

An enterprise using a market exchange platform like Commerce One's Global Trading Web for indirect goods can pre-register information about authorization levels for specific employees and corporate bank accounts to be used for payment on purchases. When a purchase is actually made, the purchaser clicks the CitiConnect icon in Commerce One's platform. At this point, an XML based message containing payment instructions is automatically assembled, including the amount involved, the identity of the purchaser, the identity of the supplier, the bank that funds are to be withdrawn from and the bank that the funds are to be transferred to, along with details regarding timing of payment. The XML message is then automatically routed according to pre-defined rules to appropriate specialized settlement networks. This approach significantly increases convenience for the purchaser by knitting together a variety of pre-existing applications using web services standards and protocols.

The benefits for both buyers and sellers are compelling -- sellers reduce the time required for settlement by 20-40% and both buyers and sellers reduce settlement costs by 50-60%. Citibank benefits by taking operational capability and building it into a new service line, reaching a broader range of customers through platforms like Commerce One. Commerce One also benefits by providing the users in its Global Trading Web with access to value added services from highly credible providers like Citibank and eliminating the need to develop these services itself. Instead, it can focus on signing up more buyers and sellers, using value added services like Citibank's as an additional draw for buyers and sellers to move more of their transaction activity online. Commerce One understands the value of this relationship and is aggressively recruiting other companies to

offer value added services like integrated logistics services on its market exchange platform.

The relationship between Citibank and Commerce One illustrates a broader pattern that will drive adoption of the web services architecture and generate revenue growth opportunities for many traditional enterprises. All providers of inter-enterprise applications (e.g., private trading exchanges, procurement services, supply chain management and e-commerce selling platforms) recognize the need to add functionality rapidly, for example providing value added services like invoice presentment and payment, product inspection, shipping and insurance. They have two choices - develop this functionality internally or source it from specialized providers. Many of these application providers are choosing the latter path. In this way, they can get to market more quickly and focus their own scarce resources on customer acquisition. The web services architecture significantly enhances the ability to collaborate with specialized third party providers in a much more flexible and low cost way than would otherwise be possible.

Traditional enterprises have an advantage here. Faced with the choice of sourcing a web service from a well-known enterprise with a strong track record or from a small start-up with issues about long-term viability, many companies are likely to prefer the former source.

As traditional enterprises begin to recognize the opportunity to grow revenue by making specialized capabilities available to other companies using the web services architecture, the importance of the service grid foundation will become more apparent. Web services delivered by traditional enterprises in areas like invoicing, payment and logistics are not marginal to their customers – they are mission critical. Without the robust functionality in areas like security, reliability and performance auditing that only specialized service grid utilities can provide, few enterprises will be willing to offer, much less subscribe to, these mission critical services.

As traditional enterprises move more aggressively to exploit revenue growth opportunities created by the web services architecture, an interesting dynamic will begin to play out. A complex ecosystem of web services participants (which we call an “eco-web”) will emerge and evolve.

Increasingly, the distinction between web services providers and web services users will become less relevant. Enterprises will focus on providing web services to others in areas where they have distinctive capability, while at the same time using web services from others in areas where they do not have distinctive capability. As time goes on, the location of distinctive capability (i.e., whether it is inside the enterprise or outside the enterprise) will become less relevant than the ability to discover and orchestrate distinctive capabilities across enterprises

to deliver more value to customers. In the process, the enterprise will be turned "inside out" – with more of its internal capabilities visible and accessible to others and more external capabilities visible and accessible within each enterprise.

Build leveraged growth platforms

This dynamic sets the stage for much more leveraged growth strategies. On the one hand, companies will have the flexibility to focus much more tightly on the activities where they have real distinctive skill. In another article, John Hagel and Marc Singer describe taxonomy of three more focused business models that are likely to emerge – customer relationship businesses, infrastructure management businesses and product innovation and commercialization businesses. (Fn. John Hagel and Marc Singer, "Unbundling the Corporation", Harvard Business Review, March-April 1999).

This unbundling and tighter focus on a particular business model will only be a prelude to much more rapid growth. These focused businesses will use the flexibility and collaboration capabilities of web services architectures to mobilize a much broader range of resources to reach a broader range of customers. While traditional mergers and acquisitions will drive some of this growth, much more leveraged approaches to growth will now be feasible – why acquire assets if it is simpler, faster and more cost effective to simply access them on an as needed basis?

Freed from the straitjacket of existing enterprise-centric technology architectures, aggressive business managers will be able to create much more economic value through orchestration of resources, rather than ownership of resources. Perhaps the most significant opportunity will be for managers to harness this orchestration capability to move well beyond the business process reengineering focus of the last two decades and reengineer processes across entire industries and markets. We are likely to see an increasing bifurcation between these meta-integrators and more specialized resource providers – with different organizational forms appropriate for these two types of businesses

Of course, exploiting these growth opportunities will require much more than a new technology architecture. Very different organizational capabilities must be developed, including new skills, performance measurement and reward systems and organizational learning approaches. Even more fundamentally, business managers will need to adopt a new mindset – embracing shaping opportunities by helping to define and deploy standards, rather than simply hoping to adapt to a rapidly changing environment.

However, none of this will be possible within the confines of existing enterprise architectures. To capture the largest business value creation opportunities on the horizon, management will need to "break on through to the other side" by embracing new technology architectures – and associated organizational forms - in a sequenced manner, driven by a clear understanding of, and demand for, real business value at each stage.

Sidebar: Key components of the service grid

The service grid promises to be a rich spawning ground for specialized utilities to provide key services for the web services architecture:

Transport management utilities – include robust messaging utilities to facilitate reliable and flexible communication across distributed application services as well as orchestration utilities helping to orchestrate the composition of a set of application services in a trusted environment according to predefined security policies.

Resource knowledge management utilities – include service directories, brokers and common registries necessary to describe available services, determine appropriate ways of interacting with services and help to create awareness of user and application context. They also include specialized services to convert data from one format into another.

Service management utilities – include utilities focused on ensuring reliable provisioning of web services, managing sessions and monitoring performance to conform to Quality of Service specifications and Service Level Agreements.

Security utilities – include security services to provide authentication, authorization and accounting and to enable more granular security arrangements that can be dynamically configured.

Performance auditing and assessment utilities – include third party services to assure users of web services that they are going to obtain performance as agreed upon. In addition, performance-bonding services may be required to compensate users for damages incurred when performance falls below levels warranted.

Billing and payment utilities – include specialized billing and payment services required to aggregate charges for the use of web services and to ensure prompt

payment so that providers have an economic incentive to expand their service offerings

Sidebar: Why ASP's are experiencing trouble

Specialized third party providers of application services will represent a compelling value proposition for enterprises over time. However, most first generation Application Software Providers (ASP's) are struggling because they are really RSP's (Repurposed Software Providers), taking applications developed in the current enterprise technology architecture and simply moving them outside the firewall. As a result, they do not exploit the potential benefits of the web services architecture and at the same time introduce additional concerns

- No real improvement in application functionality – in fact, complexity of large enterprise applications is often a liability for smaller enterprises
- Tight coupling of enterprise applications tends to reduce ability to adapt rapidly to changing business needs
- Point to point connectivity approaches increase the expense and complexity involved in collaborating across a broad range of business partners
- By delivering these applications from outside the firewall, these ASP's introduce performance concerns around security, reliability and availability, as well as concerns about the longevity of the service providers

---

Sidebar: Web services implications for the IT organization

CIO's will need to anticipate the organizational impact of a web services architecture along the following dimensions:

- The IT organization will need to move in two dimensions simultaneously – outsourcing a growing range of IT activities as credible and reliable providers of superior services become available while at the same time designing and offering IT- based services to other companies where the IT organization has superior capabilities. The CIO will increasingly become a strategist and business entrepreneur – rigorously assessing areas of competitive advantage and

focusing internal resources on building new IT-based service businesses.

- Web services architectures require a new set of technology skills in areas such as enterprise application architecture, enterprise application integration, application development, security, and IT operations. Technologists with these web services skills come from a very different background and culture relative to traditional enterprise technologists yet both sets of skills must be effectively integrated into the IT organization. The CIO will increasingly become a knowledge broker, integrating diverse skill sets that reside both within the enterprise and outside the enterprise.

- IT operations and performance will increasingly depend on effective integration of resources of multiple third parties, requiring deeper skill sets in structuring and managing third party relationships. The CIO will increasingly become an external relationship manager – defining and managing business relationships with a growing range of other companies.

- Web services offer the potential to integrate business operations across multiple enterprises, but only if broad-based standards like XML become a foundation for much more focused communities of meaning among the enterprises that need to collaborate. IT organizations will need to become much more active in shaping and even defining standards to accelerate the formation of these communities of meaning. The CIO will need to move from a leadership style based on command and control to one based on persuasion and influence.

Sidebar: Questions the senior manager should be asking about web services

After reading this paper, here are five questions that senior managers should use to engage their organizations on web services

- Do we have a shared vision among our senior management of the long-term (i.e., 5 – 10 years) business implications of this new technology architecture?

- Does our enterprise have a migration plan to this new technology architecture that realistically balances the state of development of the technology with a clear understanding of the areas of highest business impact?

- Are we exerting sufficient leadership in shaping both the functionality offered by providers of web services technology (e.g., defining performance levels required for mission critical applications) and the standards that we will need to collaborate more effectively with our business partners?

- Are we moving sufficiently aggressively in the near-term accelerate learning and to exploit the business opportunities created by this technology architecture in terms of

  - More effectively integrating with the operating processes of our business partners to lower cost and/or add value?

  - Outsourcing business activities where we are not world-class?

  - Designing business services in areas where we do have world-class capabilities and offering these to third parties?

- Do we have a clear understanding of the organizational obstacles that prevent us from moving even more aggressively to exploit the business value of this technology architecture and do we have initiatives under way to overcome these obstacles?

John Hagel III is an independent management consultant who work focuses on the intersection of business strategy and technology. His most recent book, Out of the Box: Strategies for Achieving Profits Today and Growth Tomorrow, was published by Harvard Business School Press in October 2002. He can be reached through his website www.johnhagel.com or by e-mail at john@johnhagel.com

John Seely Brown was the director of Xerox PARC until 2000. He continues his personal research into digital culture, learning and Web services. His most recent book (co-authored with Paul Duguid) is The Social Life of Information. He can be reached by e-mail at jsb@parc.com.

