2003

A Framework to assess the value of web services

Robert Campbell

Rochester Institute of Technology

B. Thomas Golisano College
of
Computing and Information Sciences

Master of Science in Information Technology

Thesis Approval Form

Student Name: Robert G. Campbell

Project Title: A Framework to Assess the Value of Web Services

Thesis Committee

Name	Signature	Date
Prof. Jeffrey A. Lasky Chair		12/9/03
Prof. Michael W. Axelrod Committee Member		12/9/03
Prof. Steven Zilora Committee Member		12/9/03

Thesis Reproduction Permission Form

Rochester Institute of Technology

B. Thomas Golisano College
of
Computing and Information Sciences

Master of Science in Information Technology

A Framework to Assess the Value of Web Services

Date: 12/11/03 Signature of Author: ______________

ROCHESTER INSTITUTE OF TECHNOLOGY

A Framework to Assess the Value of Web Services

Thesis

December 12, 2003

Abstract: Large organizations often begin to adopt new software technologies prior to establishing appropriate value frameworks. This approach may produce sub-optimal investment decisions and technology adoption rates, and introduce excessive risk. In this thesis, a value-based framework is developed for assessing the impact of Web Services technology investments on business systems development. The value factors included in the framework are data management, application development and deployment, system integration, and response time to market opportunities.

Acknowledgements

The author is grateful to Professor Jeffrey Lasky for his contributions to the development of this thesis. Also, the author would like to thank Professors Michael Axelrod and Steven Zilora for their thoughtful reviews and feedback.

Table of Contents

Figures

Tables

1 Organization & Definitions

1.1 Organization

This thesis is organized as follows:

Chapter 2 provides a description of the technical and business opportunity that Web Services affords enterprises.

Chapter 3 contains a survey of relevant literature used in the development of this thesis.

Chapter 4 provides a background on Web Services concepts and technologies to establish a technical framework for this analysis.

Chapter 5 introduces the methodology used in this thesis to examine the value contribution of Web Services to the enterprise.

Chapter 6 contains the detailed analysis and assessment of cost benefits and revenue opportunities associated with implementing Web Services. Impacts to four key areas are examined: data management, system integration, application development and deployment, and response time.

Chapter 7 summarizes current obstacles to realizing the full potential value to the enterprise of implementing Web Services.

Chapter 8 presents the conclusions drawn from the research and analysis presented in this thesis.

Appendix A provides a summary of the meaning of acronyms used in this thesis.

Appendix B provides a list of general resources for further information on Web Services.

The **Bibliography** provides a complete list of literature and articles referenced in the completion of this thesis.

1.2 Definitions

Real options – The Real Options valuation methodology accounts for the value of flexibility to adapt decisions in response to unexpected market developments and contingent on the arrival of information. Real options are based on financial options theory (Copeland & Antikarov, 2001).

Services grid – The composition of technology components and business processes that affords the opportunity to create economic value by responding to unanticipated business opportunities (Hagel & Brown, 2002b).

Value framework – A framework within which the business value (cost benefits and revenue opportunities) of a specific investment strategy can be examined (Boehm, 2003).

Web Services and Service-Oriented Architecture (SOA) – "Web Services is a technology that allows applications to communicate with each other in a platform and programming language-independent manner. A Web Service is a software interface that describes a collection of operations that can be accessed over the network through standardized XML messaging. It uses protocols based on the XML language to describe an operation to execute or data to exchange with another Web service. A group of Web Services interacting together in this manner defines a particular Web Service application in a Service-Oriented Architecture (SOA)." (IBM, 2003)

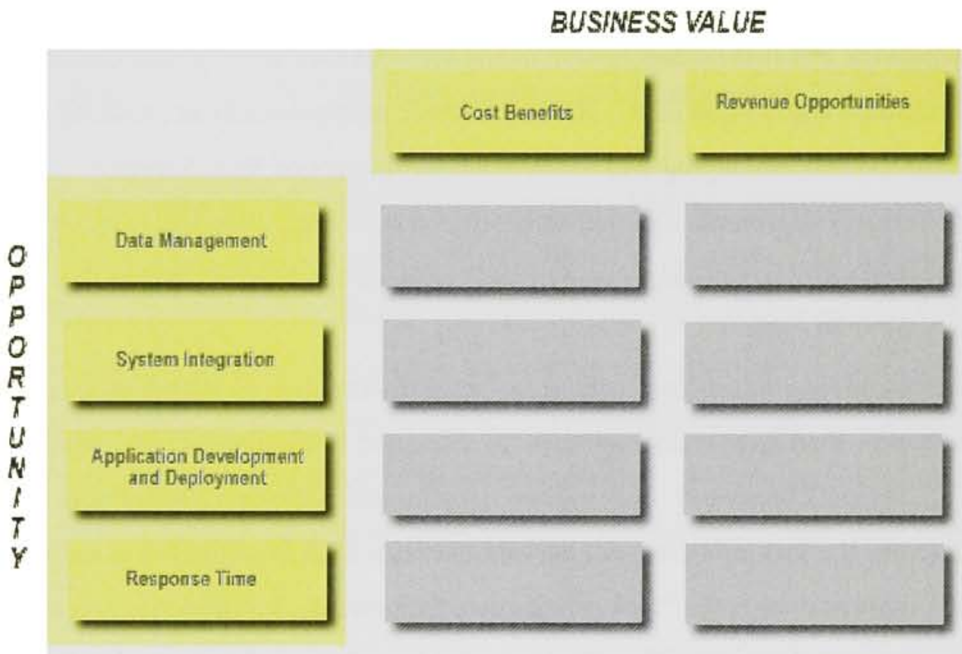
2 The Potential

Web Services technologies comprise a service interoperation architecture where components provide a real-time description of the services they provide and dynamically connect to form a distributed application (IDC, 2003b). Service architectures use standard protocols for content description and messaging and so the potential impact of Web Services is vast. Many believe Web Services represents the next technical wave of architecture drivers, and will impact future IT infrastructures on the same scale as did client/server architectures in the 1980s and 1990s.

In the corporate community, technology investments such as Web Services ought to be examined by senior managers with respect to the expected value that each investment will yield. However, in *Software Economics: A Roadmap*, Boehm and Sullivan (2000) argue that the primary reason for the lack of a clear connection between such investments in software and systems and value creation is the "lack of adequate frameworks for modeling, measuring and analyzing" investment decisions. Therefore, in order to make an adequate value-based assessment of investment in Web Services technologies, we must first establish the framework within which that assessment can be conducted.

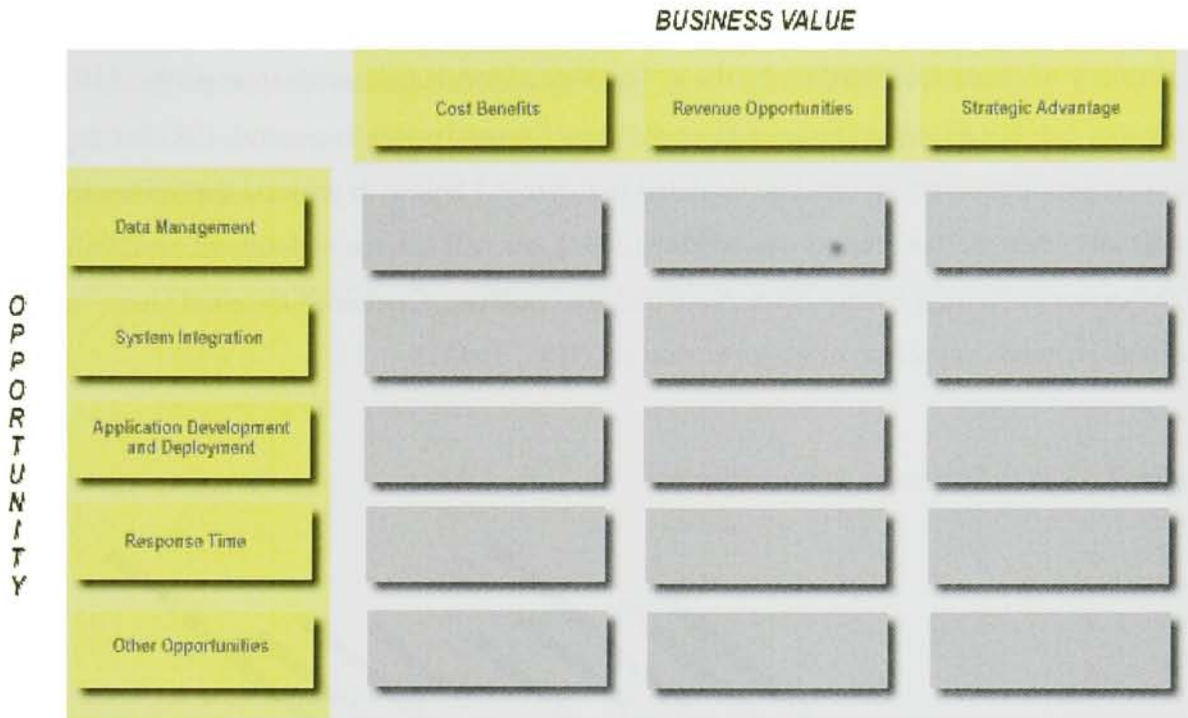
Reducing overall costs has traditionally been a key factor in determining the value of IT investments and should be included in the framework. Cost reduction alone however, is an incomplete measure of the value of software investments (Boehm & Sullivan, 2000). All investments, including IT investments, should be considered in the broader context of business value creation. A more complete approach would be to evaluate IT investments such as Web Services in the context of both cost savings as well as revenue opportunities that these investments make possible. Figure 2.1 is the proposed value framework within which a more complete value-based assessment of Web Services can be made. (A more detailed description of this framework is provided in Section 5.)

Figure 2.1 - The Value Framework



While taking into account both revenue opportunities as well as cost benefits represents a more complete value framework than assessing cost benefits alone, we acknowledge that this framework does not address other components of business value that could be considered. For example, one component of business value that is not addressed by this framework is the value of making IT investments to create strategic advantage (Luftman, Lewis & Oldach, 1993). Not all investments in technology are intended to reduce costs or generate immediate revenue. Some are intended to create strategic advantages that drive future business value. Furthermore, a more complete value framework could be presented by addressing more areas of opportunity, of which there are many. Therefore, he following diagram may represent a more complete value framework for the evaluation of IT investments such as Web Services technologies.

Figure 2.2 – A More Complete Value Framework



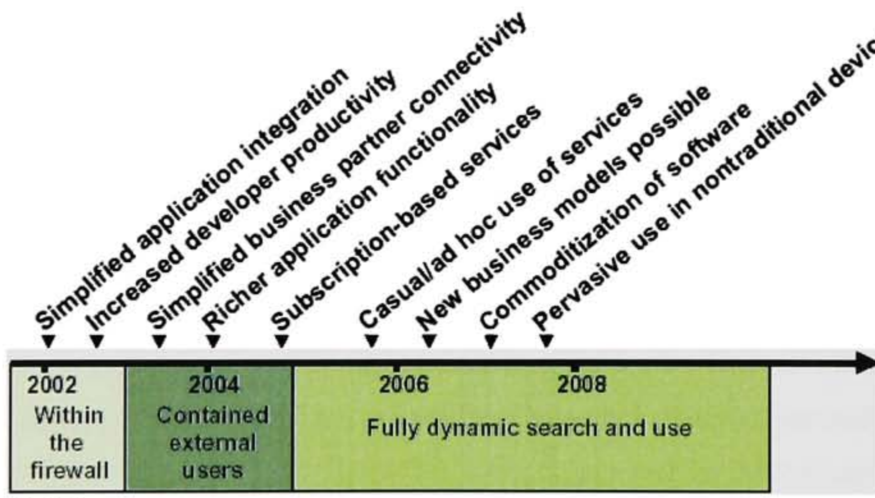
Nevertheless, the scope of this thesis will be limited to searching for cost benefits and revenue opportunities in the areas of data management, system integration, application development and deployment, and response time; i.e. we will be using the framework as proposed in Figure 2.1.

Using the framework, we are able to create explicit links between technology investments and business value (Luftman et al., 1993). We will show that Web Services technologies afford enterprises the prospect of reducing overall IT costs by increasing efficiency in these areas of opportunity. Additionally, we will show that Web Services will create value, i.e., enable new revenue opportunities, by improving an enterprise's ability to respond to emerging customer needs in the marketplace.

The objective of this thesis is to evaluate the potential value contributions of Web Services within this framework and to explicitly identify the resulting cost benefits and revenue opportunities. This integration of value considerations into software investment decisions enables a more comprehensive recognition of the potential value contributions of Web Services (Boehm & Guo, 2003).

(At the time of this writing, the eventual level of acceptance and adoption of Web Services technologies remains uncertain. The rate at which these technologies are adopted will depend greatly on a number of factors including the maturation of proposed standards, the availability of commercialized tools, the availability of individuals with appropriate skill levels, economic conditions and patterns of IT investment for enterprises in general. IDC has proposed the Web Services adoption timeline depicted in figure 2.3 below. It projects that, given the conditions when the IDC report was written (2002), we will not see widespread availability of Web Services offerings before 2005. Note that IDC defines "Contained External Users" as business partners, suppliers, or major customers (IDC, 2002).)

Figure 2.3 - Web Services adoption timeline



Source: IDC, 2002

The expected benefits of Web Services described herein, and the associated value-based contributions to the enterprise are dependent on broad, industry-wide acceptance of Web Services technologies and standards across enterprises. With that in mind, the following summarizes the hypothesis being addressed in this thesis.

Hypothesis

Business justification for investing in Web Services technologies can be based on the expectation that these technologies will contribute business value to the enterprise in the form of identifiable cost benefits and revenue opportunities.

3 Literature Review

3.1 The Relevance of Value

This thesis focuses on assessing the value of IT investments, and specifically the value of investments in Web Services technologies. This presumes that we can define value as it pertains to Web Services, and that the notion of value is relevant to such investments.

In his article *Fundamentals of Value*, Sawhney (2003) asserts that the definition of value should be considered to be contextual. The value of technology is not independent from the context in which it is used. Without a thorough understanding of the business context and implications of a technology investment, there is a risk of creating value propositions that are not relevant for customers. When the business context is well understood, value can be defined, created and delivered to customers. In addition Sawhney argues that the value of IT investments should be defined not in terms of features and functionality but rather in terms of what it can do for customers. "Value," he says "is defined by those…who pay for it".

We define value in terms of revenue opportunities and cost benefits as shown in the framework offered in Section 2. This framework provides the context for linking business value (and value propositions for the customer) to IT infrastructures. Furthermore, by defining value in part as revenue opportunities, we build on Sawhney's assertion that (Web Services) investments should be directly linked to the needs of customers and that such investments result in new revenue opportunities.

Britt, of the IBM Institute for Business Value, says, "Consultants have long proclaimed the need for strategic alignment between business and IT: Set your business strategy, and then determine how technology can help. Unfortunately, traditional alignment approaches invite risk and leave opportunities untapped. Higher returns can be achieved through a higher degree of strategic alignment — the fusion of business and IT". Like Sawhney, Britt argues that IT investment can only create value to the extent that it is carefully aligned with the enterprise's business strategy (Britt, 2002).

This focus on value is a recurring theme in many leading publications focusing on business and technology leadership. And creation of value is commonly linked to making certain that each and every investment that an enterprise makes supports both the business strategy of the enterprise and customer needs. "Value is not a synonym for profit", suggests Moneypenny (2003). It can, however, be measured in that way. Value is more. It is multi-dimensional. Value is customer defined and customer focused, but at its core is the inherent ability to meet a customer need (Sawhney, 2003).

John Hagel III and John Seeley Brown have authored a number of articles addressing the value of Web Services in supporting business objectives (Hagle & Brown, 2001; 2002a; 2002b). Most notably, in *Orchestrating Business Processes – Harnessing the Value of Web Services Technology*, they describe how IT investments in Web Services technologies can be leveraged to directly support key business processes to create customer value. Their conclusions are similar to Sawhney, Britt and others: that technology can be a significant enabler in responding to business opportunities, and that IT managers and organizations play a key role in enabling enterprises to realize economic value from Web Services investments.

John Hagle provides the most comprehensive treatment to date of the business impact of Web Services in his book *Out of the Box* (2002a). Hagle specifically addresses strategies for achieving profits through Web Services as well as enabling opportunities for future growth. He argues that Web Services technology will play a major role in creating and delivering business value by addressing three business/technical challenges facing enterprises today:

- "Distribution of centers of control" – As business processes become more complex, and the number of business partners increases, the assumption of a single point of control becomes increasingly unrealistic.
- "Diversity of technology platforms" – Given the decreasing existence of single points of control, there is a growing diversity of technology platforms that must be connected in the process of assembling business capabilities to deliver products and services.
- "Dynamic environment" – According to Hagle, business conditions are changing at an ever increasing pace and enterprises need to be able to quickly respond to emerging opportunities. Just as importantly, he argues, participants in business relationships today need to retain the flexibility to exit business relationships without incurring significant expense and without scrapping investments in technology and infrastructure.

Hagel spends a great deal of time discussing the characteristics of architectures that enable enterprises to manage these three challenges. Furthermore, he lays out his notion of a services grid, a concept that is targeted toward enabling future growth as opposed to capturing immediate revenues and profits.

Hagel's discussion of the immediate opportunities for capturing value (profits) through the pragmatic adoption of Web Services technologies directly supports the hypothesis of this thesis. He presents the path taken by early adopters such as Dell as well as their motivation and the results that have been achieved. Hagel provides extensive support for his argument that early implementations of Web Services should focus on leveraging existing technology infrastructure, incremental implementation, tangible early successes, and extensibility over time.

Barry Boehm is a leading proponent of a value-based perspective on software engineering. According to Boehm, while many IT projects are implemented successfully, the value derived is less than expected because business objectives were incorrectly anticipated (Boehm, 2003), (Boehm & Sullivan, 2000), (Boehm & Guo, 2003). Again, value generation is dependent on an alignment of technical strategy and investment with business objectives and customer needs.

IDC, working with IBM Corporation, published a white paper entitled *IBM and the Strategic Potential of Web Services: Assessing the Customer Experience*. Their findings support the importance of considering value generation with respect to IT investments. They found that "the use of Web services enhances business value by reducing the time and cost to launch applications … potentially, increasing revenue" (IDC, 2003b).

3.2 The Alignment of IT Investments with Business Strategy

To a large extent, this thesis attempts to establish a connection between Web Services technology and value creation for the enterprise. Britt, of the IBM Institute for Business Value, underscores the importance of aligning business objectives (value creation) and IT investment in *Multiplying Business Value: The Fusion of Business and Technology* (2002). Luftman et al. (1993) provide further evidence of the importance of this alignment in *Transforming the Enterprise: The Alignment of Business and Information Technology Strategies*.

3.3 The Growing Importance of Web Services and Service-Oriented Architectures

While the full impact of Web Services cannot yet be assessed, IT organizations are taking notice of these emerging standards and technologies and, in many cases, planning and initiating projects to test their usefulness. This trend is supported by an IDC study indicating that hardware, software and integration spending on Web Services is increasing (IDC, 2003a). Furthermore, there is evidence that major players in the IT domain are taking Web Services seriously. For example, IBM lists no fewer than forty emerging Web Services technologies on their developerWorks™ web site, including several implementations of SOAP and UDDI. They provide toolkits for generating WSDL and for evaluating the performance of Web Service implementations. Many of the technologies listed are intended to help early adopters of Web Service technologies to create, define, discover and integrate services. In addition, IBM researchers have recognized the overarching significance of SOA, and that the potential importance of Web Services goes far beyond the individual capabilities. In his article entitled *The Tao of eBusiness Services: The Evolution of Web Applications Into Service-Oriented Components with Web Services,* Burbeck (2002) clearly articulates the importance of approaching Web Services from the architectural perspective.

Other publications have moved beyond the initial hype of Web Services technologies to emphasize the importance of those technologies in creating value for the enterprise. Among them are (Hagel & Brown, 2001), (IDC, 2003b), and (Hagel, 2002b).

In (Hagel & Brown, 2002a) the authors focus on the economic propositions driving the implementation of Web Services, providing a description of how the pragmatic implementation of Web Services technologies can add value to the enterprise. Moschella (2003) documents those aspects of IT infrastructure that can improve customer focus. Hagel and Brown (2001) describe the issues addressed by Web Services and propose the basis for a strategic approach to introducing Web Services into IT organizations.

The March 2003 issue of ACM's *Queue Magazine* provides a detailed examination of Web Services and the opportunities they present. In this issue, Arsanjani, et al. look at the promises and compromises of Web Services (Arsanjani, Hailpern, Martin & Tarr, 2003). They assert the shift to architectures that support the dynamic interconnection of services is inevitable. The migration toward such architectures is an evolutionary step in the maturation of component-based development and integration.

3.4 WEB SERVICES TECHNOLOGIES

A number of sources provide background on Web Services technologies. Perhaps the most comprehensive starting point is the World Wide Web Consortium, better known as W3C.

W3C is responsible for the development of common protocols that promote the evolution of the Web and ensure its interoperability. XML, SOAP and WSDL are described in the work products of the W3C (www.w3c.org).

Additional background material for Web Services is found in (Barry, 2002). Barry provides a high level description of the technologies and issues in order to prepare organizations for the implementation of Web Services. Barry suggests that Web Services are going to "fundamentally change the way we build our internal systems – the information systems that support our organizations – and how our internal systems interact with external systems". An additional resource used to compile background material for Web Services is (Newcomer, 2002).

Burbank's article describes the important architectural concepts behind Web Services, and describes them as the basis of a new set of organizing principles within the enterprise (2002). Burbeck makes the important observation that the term SOA should be reserved for "architectures that focus on how services are described and organized to support their dynamic, automated discovery and use".

With respect to SOA, Hagle and Brown again have been instrumental in describing the need for thinking of Web Services in terms of the evolution of IT architectures. In their publication *Service Grids: The Missing Link in Web Services*, they assert that a distributed services architecture is necessary before Web Services technology can be widely deployed.

The *IBM Systems Journal* article, "Introduction to Web Services Architecture" provides both an introduction to important architectural concepts related to Web Services, as well as examples of business scenarios to which the technologies can be applied (Gottschalk et al.). Placing the Web Service technologies in the context of a specific business situation provides a higher level of clarity and understanding of possible applications (Gottschalk, Graham, Kreger & Snell, 2002).

3.5 Areas of Opportunity for Implementing Web Services

3.5.1 Data Management

The issues related to data management in IT organizations are described by Applegate et al. in (Applegate, McFarlan, & McKenney, 1996), (Applegate, McFarlan, & Austin, 2002), and (Applegate & Bock, 1995). The issues identified and discussed include information, organization, and control as well as the evolution and alternatives for IT information architectures. These data management issues continue to be relevant regardless of the IT technologies and infrastructures in place. Web Services may drive the next wave of architectural principles within the enterprise, but existing legacy data repositories in place must be carefully considered when creating SOA using these new technologies.

The data management analysis relies heavily on Coyle (2002). This text provides a comprehensive description of the data management issues facing larger enterprises and how XML and Web Services will address them. In addition to these text references, a number of industry reports and opinions from IDC (2003a), Forrester (2002a), and Gartner (2003) have been used.

3.5.2 System Integration

Linthicum documents many of the integration challenges faced by IT managers are discussed including disparate legacy systems, heterogeneous hardware architectures, operating systems, networking technology and proprietary applications and data formats (1999).

The primary issues pertaining to legacy software are described by Lehman and Belady (1985), and Seacord, Plakosh and Lewis (2003). These texts characterize the unique challenges of managing and upgrading legacy systems without disrupting the operation of the enterprise.

3.5.3 Application Development and Deployment

IDC asserts that Web Services will directly impact a number of key application development areas including:

- Enterprise Application Integration (EAI)
- Application construction
- Outsourcing potential
- Freedom to choose best in class capabilities for a particular solution

IDC cites specific examples of significant cost savings and increased efficiencies resulting from the deployment of Web Services platforms. The conclusions drawn by IDC with respect to Web Services applications development impact include the predictions that businesses will have more choices in how applications are constructed, business functions will be more accessible, and that dependencies on underlying technologies will be reduced (2003b).

In *Investing in Information Technology: Productivity Payoffs for U.S. Industries*, Stiroh (2001) underscores the importance of investing in information technology to improve productivity. He concludes that the acceleration of labor productivity growth in the U.S. is a direct result of IT investment.

3.5.4 Response Time

Many of the publications cited above identify one of the key benefits of Web Services and SOA as the ability to respond more rapidly to emerging, unanticipated customer needs. The core assertion is that the strategic value of Web Services is more than an improvement in technology. Rather, it is advancement in the ways in which information technology can support business objectives, and how a strategic advantage can be established in the process.

Luftman, et al. (1993) characterize the transforming impact of information technology such as Web Services in the following way: "The goal of serving the widest range of customer needs in the most cost-effective and responsive way represents a shift to new competitive strategies." The implication of Web Services deployment is that mass customization of product and service offerings is enabled by the existence of services that can be rapidly organized to meet dynamic customer needs quickly with minimum expense. Sifonis and Flynn (2001) argue that the value of Web Services is in the options that are created and the consequent ability to respond to customer needs more quickly than the competition.

Britt (2002) refers to this synergy between enabling information technologies and evolving customer needs as the creation of value through the "fusion of business and technology".

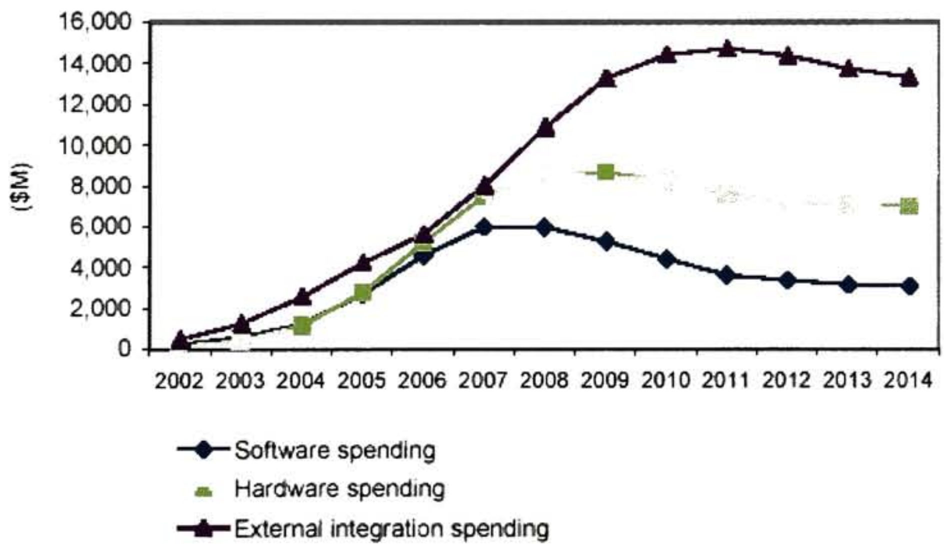
Hagel (2002a) again provides insight into the issue of enabling enterprises to respond quickly to emerging opportunities. His discussion of the services grid is helpful in understanding one way in which enterprises may prepare to capture future revenues without knowing the specific details of those opportunities ahead of time.

4 Background

4.1 WEB SERVICES EMERGENCE

This thesis asserts that business value will result from the implementation of Web Services. That value will be enabled by a number of existing and emerging technologies and standards. Figure 4.1 indicates that enterprises are indeed making substantial investments in Web Services hardware and software. IDC defines software, hardware and external integration spending in the following way. "Software includes all spending on any type of software for the purpose of the project. Hardware spending is likewise defined as spending on physical infrastructure to support the project, and system integration (professional services) spending is spending on external consultants of all kinds to implement the project" (IDC, 2003a).

Figure 4.1 - U.S. Web Services project spending by IT segment, 2002-2014



Source: IDC's Web Services Adoption Model version 12 and *Web Services Awareness Study*, March 2002

The challenge for IT organizations will be to show how these potential investments in Web Services will add value and strengthen the competitiveness of the enterprise.

Table 4.1 presents example applications that potentially will benefit from IT infrastructures that incorporate Web Services.

Table 4.1 - Range of Web Services' applicability

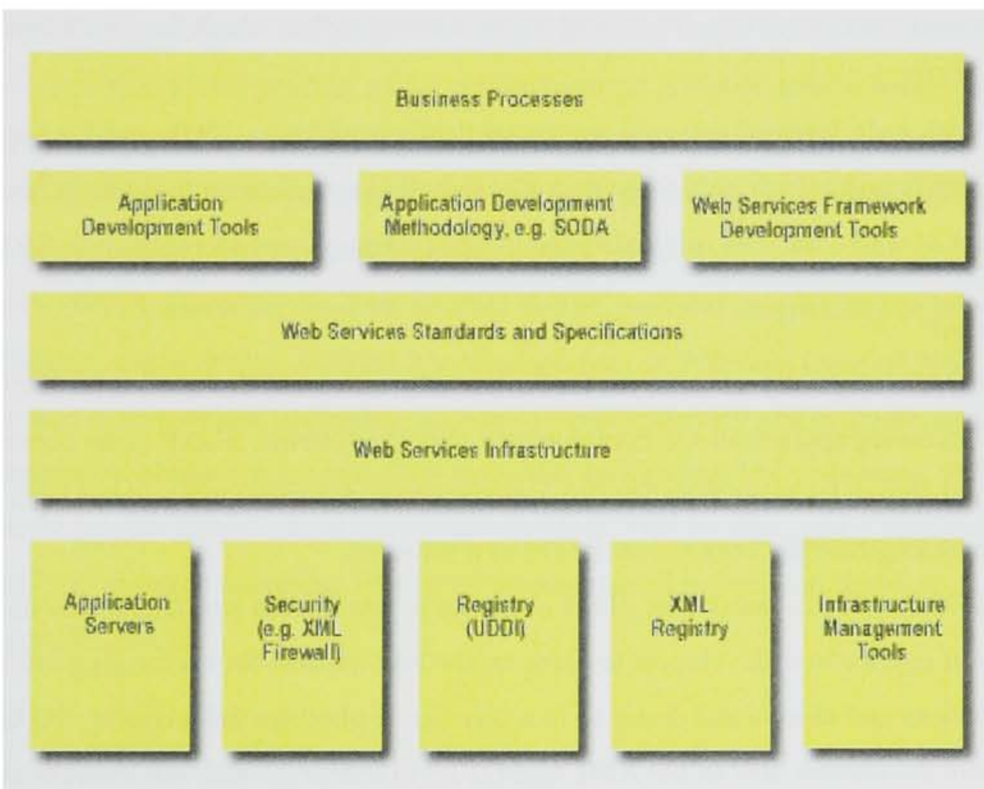
Intranet applications	**Extranet applications**	**Externally published Web Services**
• Corporate portals • Merger/acquisition integration • Communications (email, instant messaging, web conferencing) • Security and authentication • Document/information management • System management • Information aggregation • Computer-assisted training and instruction • Accounting and financial services • Distribution & product delivery • Manufacturing and engineering • Sales & marketing support	• Joint development • Collaborative partnerships • Supply chain integration • Procurement • Billing, credit • Mobile/wireless access	• eBusiness initiatives • Published/syndicated content • Customer/technical support • Computer-assisted training and instruction

The key to realizing the full potential is to recognize that Web Services presents a standards-based approach to integrating existing systems, applications such as those listed above, and data. It is not necessarily about implementing new systems from the ground up, but rather taking advantage of legacy data and capabilities in ways that allow those capabilities to be quickly and easily used by a wide variety of applications, i.e. creating options for the enterprise.

In the past, data integration has been an especially difficult challenge. The use of XML and other industry standard technologies provides a framework for universal data integration. The standards upon which Web Services is based attack the problem of data exchange and interoperability while preserving platform independence. In addition, XML-based messaging makes it possible to deliver application independent data in real time. This will enable the rapid introduction of new services that may have otherwise have taken years to develop because of the labor-intensive nature of developing and testing new interfaces.

Web Services is implemented in the context of a SOA. A SOA is composed of loosely coupled collections of services, often organized around business capabilities rather than technical functions. Figure 4.2 illustrates the high-level components of a SOA (Hagel & Brown, 2002b).

Figure 4.2 - SOA components



The following sections provide a brief description of the core Web Services technologies.

4.2 XML

Extensible Markup Language (XML) is the catalyst driving the adoption of Web Services. XML allows data to be defined in a way that separates information about content from information about presentation. This approach addresses the primary flaw of Hypertext Markup Language (HTML), which combines data and presentation into a single markup language (Newcomer, 2002).

The significant impact of XML is not rooted in the expectation that all data everywhere will be stored in XML format, but rather that given a standards-based data transformation capability, legacy data from existing repositories can be made available to XML-enabled applications. The transformation is required only once. Middleware applications often require multiple data transformations depending on the context within which the data would be used (Coyle, 2002).

With well designed schemas and semantically useful tags, XML enables applications to easily identify and extract only those data elements that are useful in the current context regardless of where the data originated and in what format it was stored (Coyle, 2002).

4.3 SOAP

SOAP (originally an acronym for Simple Object Access Protocol) is an XML-based protocol allowing applications to communicate over the Internet. SOAP is the communication protocol of choice for most Web Service applications and frameworks and has been proposed to the Internet Engineering Task Force (IETF) as a standard.

SOAP is XML-based. One of the obstacles to application interoperability has been the inability of applications developers to agree on the best method for exchanging information about services and objects and doing so in a way that is platform independent. SOAP defines for component developers how to use XML and HTTP to exchange such information. And since HTTP is ubiquitous, and XML is becoming so, it provides a glue between dissimilar components on heterogeneous networks (Newcomer, 2002).

4.4 UDDI

Universal Description, Discovery and Integration (UDDI) is an XML-based directory facility that allows service providers to advertise their capabilities to applications searching for services (Newcomer, 2002). Initially, it was thought that UDDI would provide a grand opportunity for eBusiness directories that would act as the basis of the "frictionless economy", providing rapid identification and invocation of Web Services across the Internet.

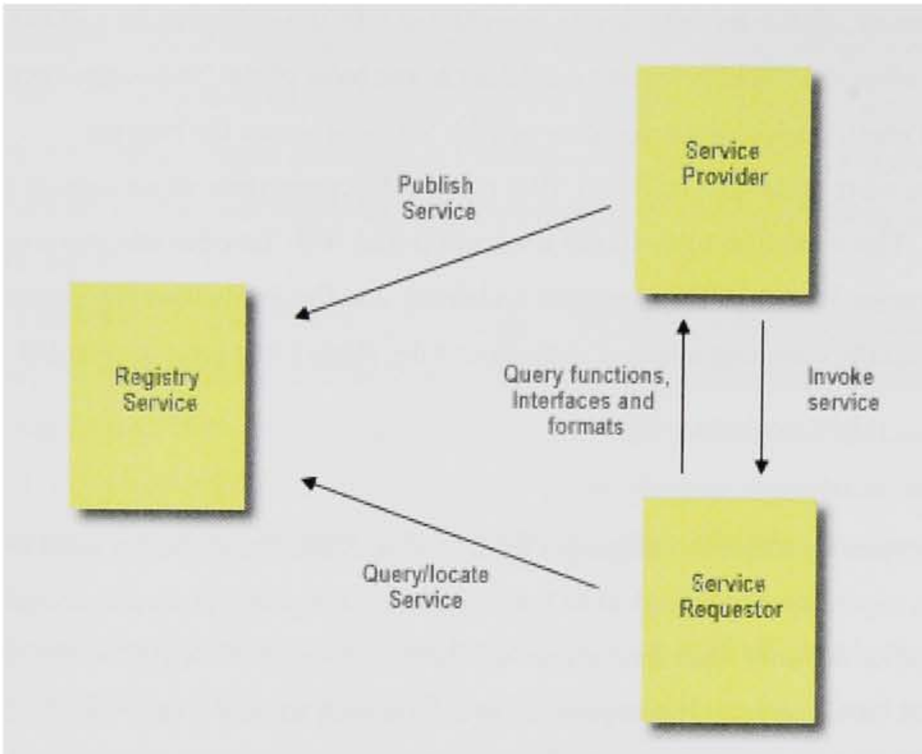
Now, the more pragmatic role for UDDI is to be the cornerstone of services oriented IT infrastructures. This evolution supports the notion of initial Web Services adoption occurring within the enterprise. Finally, UDDI provides scalability and flexibility, two key characteristics in meeting the rapidly changing needs of applications developers in a service-oriented environment.

4.5 WSDL

Web Services Description Language (WSDL) is an XML-based facility used to describe a Web Service's capabilities independent of the communication protocol used to access those services. It specifies not only what services are available, but also how WSDL works in conjunction with UDDI to provide comprehensive information about the capabilities of available Web Services on the Internet (or Intranet) (Burbeck, 2002).

In a SOA, WSDL meets the need for an XML-based, protocol independent mechanism for easily publishing, and updating, services in a standard format. Potential users of Web Services advertised in this yellow pages-like fashion can easily identify and compare available services and determine which may be best for a particular application. For consumers of Web Services, dynamic service identification and update are simplified by eliminating details of access protocols and message encoding. Figure 4.3 summarizes the core Web Services operations (Burbeck, 2002).

Figure 4.3 - Core Web Services operations



4.6 APPLICABLE STANDARDS

The existence of proposed standards, specifications and methodologies that provide the basis for a Web Services framework is instrumental to increasing the likelihood that applications will meet interoperability expectations. While progress is being made, the standards underlying Web Services are still evolving, and so there is still the potential for vendor specific implementation and fragmentation that could undermine the primary objective, indeed the promise of Web Services, of enabling a truly interoperable framework of disparate information producers and consumers (Gottschalk et al., 2002).

In addition to those standards mentioned above, Extensible Style sheet Language (XSL) is critical to enabling the development of Web Services applications. XSL provides consistent grammars for defining the ways in which data is transformed (Newcomer, 2002).

4.7 APPLICATION FRAMEWORKS

The distinction between Web Services applications and Web Services frameworks is important for full appreciation of Web Service's potential to add value for the enterprise. Web

Services applications are developed within the context of a framework. They are software applications, products or services that make data and processing capability available to other applications. These applications can be considered independent building blocks advertising and providing services or capabilities to other applications on a network. These applications can be combined in virtually limitless combinations to provide end users a unique information product or service. A Web Services framework, on the other hand, consists of standards, specifications and methodologies for building such interoperable applications. J2EE and .NET are two of the leading application frameworks for developing Web Services applications. These frameworks can be used to create and publish Web Services (Forrester, 2002b).

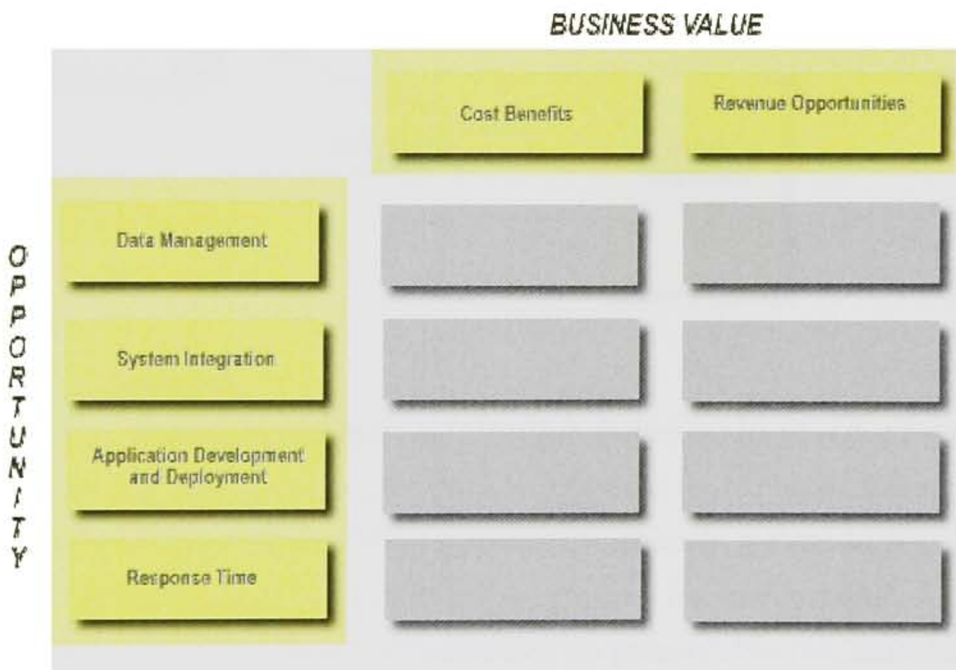
5 Methodology

5.1 Approach

5.1.1 The framework

As mentioned above, the following diagram describes the framework within which we will evaluate the potential value contributions of Web Services.

Figure 5.1 – The Value Framework



5.1.2 Identifying value within the framework

For each of the above four identified areas of opportunity – data management, system integration, application development and deployment, and response time - the benefits of Web Services implementations will be established by showing how such implementations result in one or both of the following:

- **Cost benefits** in the form of increased efficiency in the use of IT resources for software development, and

- **Revenue opportunities** that are realized as a result of implementing Web Services technologies

5.1.2.1 Cost Benefits

Cost efficiencies can result from lower capital investments, lower ongoing support costs, or reduced labor costs. While cost efficiencies alone are insufficient to establish a comprehensive value framework within which to make investment decisions, such decisions are often driven in part by the expectation that cost efficiencies will result (Boehm & Guo, 2003). Investments in IT technologies such as Web Services are no exception. In this thesis we will identify and describe the cost efficiencies resulting from the adoption of Web Services technologies.

Cost benefits for each of the four areas of opportunity are summarized in table format in sections 6.1.2, 6.2.2, 6.3.2 and 6.4.2.

5.1.2.2 Revenue Opportunities

Revenue opportunities resulting from implementation of Web Services technologies are manifested in at least two forms. First, by delivering products and services in the target markets where a firm already competes may generate new revenues. These products and services may be of better quality or contain enhanced functionality, but they largely represent delivery of products to existing customers using (and extending) existing business processes.

Second, new revenue opportunities are afforded firms that have the ability to respond to emerging business opportunities. This is especially true with those firms whose business models are heavily dependent on rapidly changing technologies such as Internet technologies. Being first to market allows a firm to capture market share and establish itself as the market leader. Web Services may open up opportunities for rapid delivery of new, unanticipated information-based products and services, creating new sources of revenue outside of a firm's existing target markets.

New revenue opportunities identified for each of the four areas of opportunity are also summarized in table format in sections 6.1.2, 6.2.2, 6.3.2 and 6.4.2.

5.1.3 Opportunities

The vertical axis of this framework identifies four areas of opportunity to be examined in detail with respect to Web Services implementation. They are:

- **Data management**
- **System integration**
- **Application development & deployment**
- **Response time**

These four are selected because they represent important value drivers of competitive advantage for firms in today's markets, regardless of the product or service they may provide.

5.1.3.1 Data Management

Data Management is selected as one dimension of the value framework because lack of industry standards and lack of enterprise-wide data definitions have resulted in pervasive inefficiencies. Improvements in Data Management techniques have the potential to return value to the enterprise both in the form of reduced costs as well as revenue opportunities.

5.1.3.2 System Integration

Web Services may be viewed as an advance in system integration. The value is realized when they are integrated with other services in a unique way to provide a solution to solve a specific business problem. System Integration has been selected as a dimension within the value framework because of the historical complexity of integrating large systems and solutions, and the potential for Web Services to simplify this process.

5.1.3.3 Application Development & Deployment

Application development and deployment has traditionally been slow and inefficient as new IT technology is introduced, consuming significant resources in the process. Reducing the resources required to develop and deploy new applications represents a significant opportunity for enterprises.

5.1.3.4 Response Time

The value of IT advancements is often derived not from the technology but rather from the business possibilities that result (Masud, 2002). With Web Services enterprises can focus on assembling services and solutions from a technology agnostic viewpoint, worrying only about the orchestration that makes it possible. For this reason, response time (or, the reduction thereof) is selected as the last dimension of the value framework.

Table 5.1 describes in more detail the opportunities that make up the framework within which this thesis will proceed.

Table 5.1 - Detail of issues to be evaluated

Data Management	**System Integration**	**Application Development & Deployment**	**Response Time**
• *Data accessibility* • *Data transformation* • *Data semantics* • *Content management*	• *Service-Oriented Architectures (SOA)* • *Platform independence* • *Web Services and distributed computing* • *Just-in-time integration* • *Vendor-supplied web services*	• *Application development* • *Application deployment* • *Use of "best in class" capabilities* • *Reduced it resource requirements* • *Development outsourcing*	• *Reduced time-to-market in existing market segments* • *New business models and emerging market opportunities* • *Reduced barriers to entry*

Data Management	System Integration	Application Development & Deployment	Response Time
	frameworks • *Web Services management* • *Legacy systems*	• *Service aggregation within the enterprise*	• *Leveraging existing intellectual assets* • *Integration of mergers and acquisitions* • *Response to regulatory requirements* • *Compliance with vertical and horizontal industry standards* • *Focus on business processes*

6 Analysis

6.1 DATA MANAGEMENT

Enterprises have been faced with complex data management issues for decades. Lack of industry standards and lack of enterprise-wide data definitions have resulted in pervasive inefficiencies. Changing existing data management practices, however, can have significant cost implications and wide-ranging impacts across an enterprise. Therefore, such investments are not made without a clear understanding of the potential benefits to be realized.

With respect to data management, the expected benefits of investing in Web Services technologies, especially the use of XML, include improved data exchange, reduced maintenance costs and reduced development time. The following section provides a detailed analysis of how these benefits and others applicable to data management will be realized by investing in Web Services technologies.

For the purpose of this analysis, data management has been further broken down into the following sub-categories:

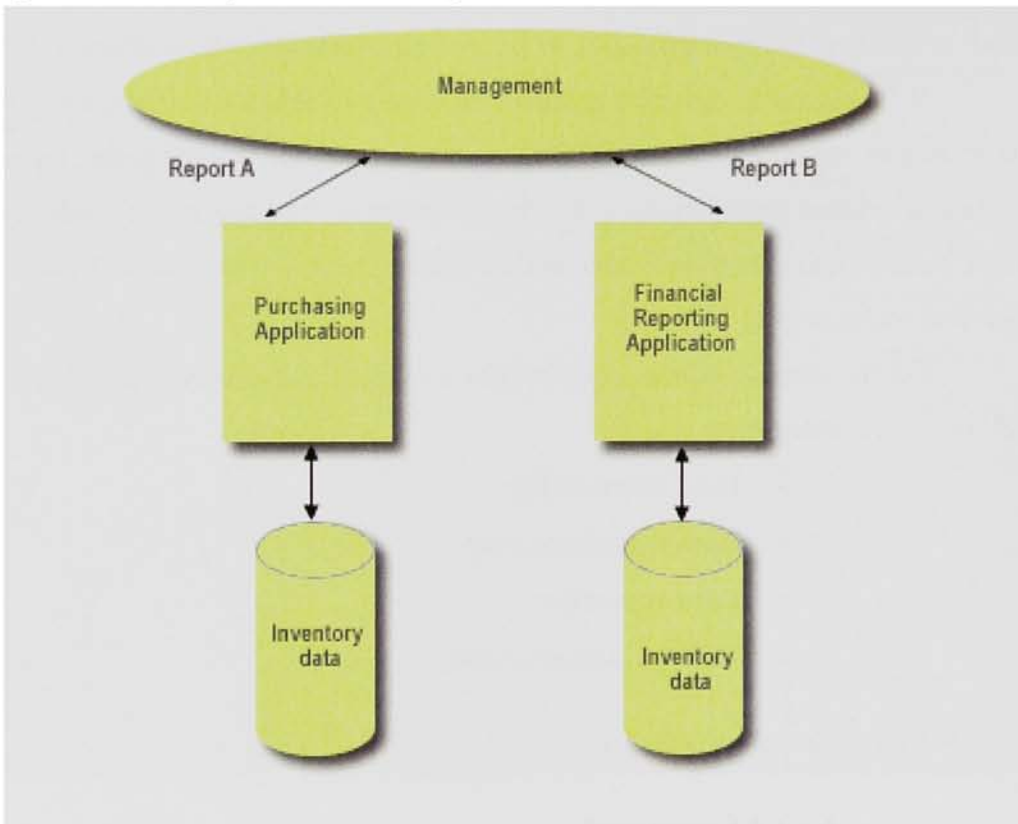
- **Data accessibility**
- **Data transformation**
- **Data semantics**
- **Content management**

6.1.1 Technical analysis

6.1.1.1 Background

Enterprises have been expanding and evolving the approach to building and maintaining data repositories since the early days of computing. Initially, data was stored in computer-accessible files and accessed by specialized software applications. The applications and data files were tightly integrated so that changes to one often required changes to the other. As multiple applications accessed the same files, the complexity of coordinating changes increased quickly. Often, to avoid the ripple effects of such changes, data was replicated in multiple locations and formats. Distributing data files and formats, however, only temporarily eliminated interdependencies. Managing a single data item in multiple locations resulted in redundancy, inconsistency, and ultimately higher maintenance costs and had to be managed carefully across the enterprise (Applegate et al., 1996; 2002). Furthermore, reports generated by applications using data from dedicated repositories could result in inconsistent reporting as shown below.

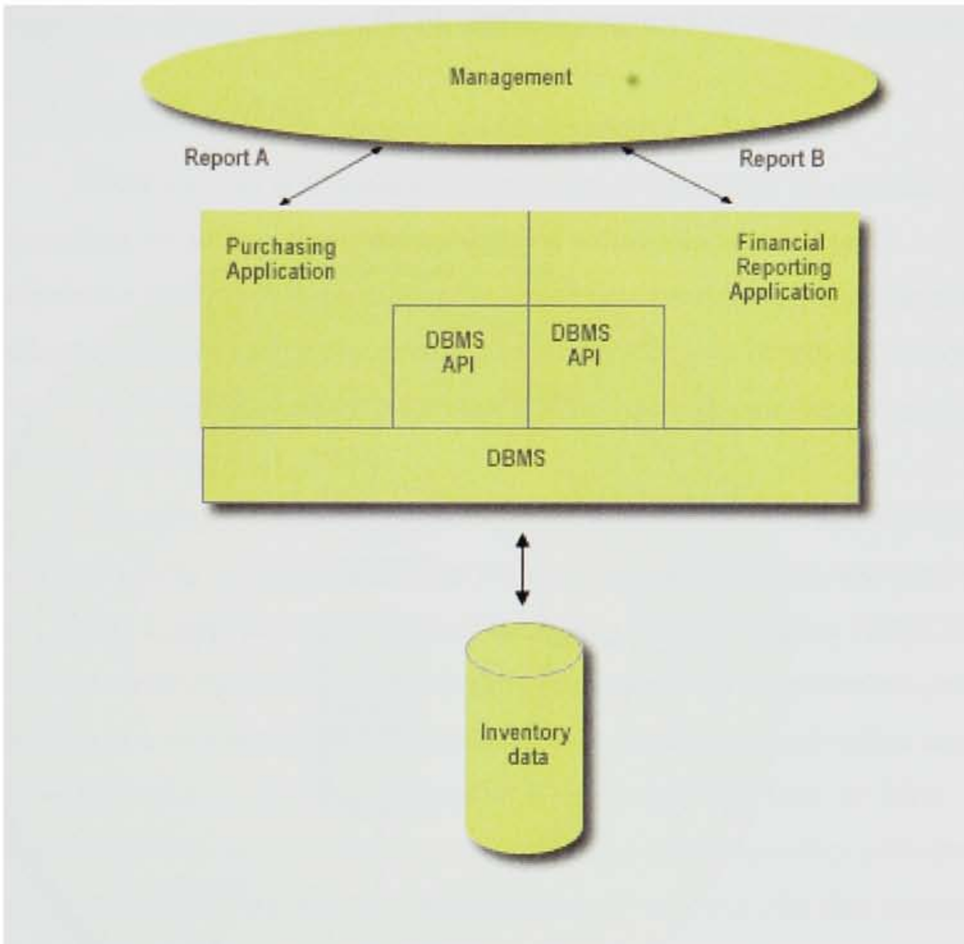
Figure 6.1 - Files systems data storage



The approach to storage of this data gradually evolved from file-based systems to Database Management Systems (DBMS). DBMS took on many functions such as data entry, update and management that were formerly performed by a specialized application or set of applications. DBMS vendors provided products that handled these roles and published a limited set of functionality through an Application Programming Interface (API). This made the data more independent from the applications requiring access. The interface between applications and DBMS, however, remained tightly integrated and highly specific to individual DBMS products.

Database systems further evolved from hierarchical, to relational, to object-oriented database systems (Applegate et al., 1996). The common thread with using DBMS to store data is that for each functional application that needs to access the data repository, a specialized and often inflexible interface must be implemented as shown below. The interfaces defined by individual vendors can vary widely from DBMS to DBMS.

Figure 6.2 - DBMS data storage

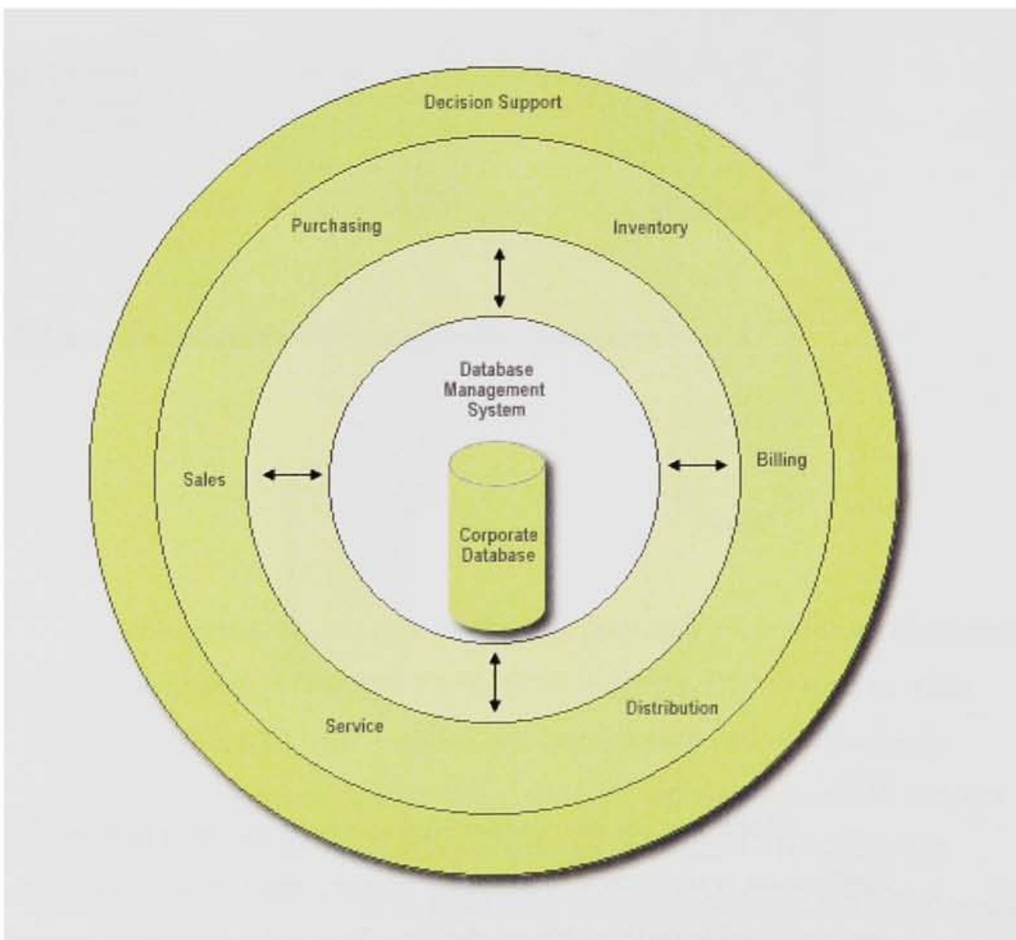


Moreover, for every repository to which access is required by an application, a separate, and probably unique, interface would need to be implemented in order to access the data. This tight coupling between applications and data repositories continues to sustain unnecessarily high maintenance costs.

An additional issue that has beset data access in information architectures in the past is that the data provided via the interface to the repository is likely to contain an inappropriate level of detail for each functional application. Rather, each application must accept records of information that have been generalized to meet the demands of multiple applications, and parse the data looking for only the pertinent information it requires. Further, each application must adapt the format of the data to a format, or type, appropriate for use in the current context.

Figure 6.3 shows a centralized enterprise data repository that eliminates the redundancy and inconsistencies possible with multiple data repositories, but retains the tightly integrated nature of the interface between applications and the repository (Applegate & Bock, 1995).

Figure 6.3 - Corporate database systems



Subsequent migration from hierarchical DMBS to relational and object-oriented DBMS has increased the flexibility of working with data from a central repository to support business processes, but has done little to alleviate the problems associated with the tightly integrated interfaces between applications and data repositories. Changes in underlying data repositories ultimately result in application maintenance (Applegate et al., 1996) (Applegate & Bock, 1995).

There is evidence that Web Services and XML will positively impact Data Management within the enterprise. The primary reason for this is the architectural advantages of separating data, and services that provide access to it, from the applications that consume that data, reducing the interdependencies that have traditionally driven up development and maintenance costs for IT organizations.

6.1.1.2 Data accessibility

Web Services will positively impact data accessibility in a number of ways. First, massive legacy data repositories will continue to exist well into the foreseeable future. Large volumes of business data reside, and will always reside, on legacy systems such as mainframe computers. The cost of moving this data will prohibit rewrite of many large scale systems that manage these legacy data repositories (Peng et al., 1998).

These legacy repositories can be made available to Web Services within the enterprise by "wrapping" them in Web Services technologies, and advertising their ability to provide certain types of data using SOAP through a WSDL defined interface (Gottschalk et al., 2002). Tightly integrated legacy applications can continue to access data through traditional APIs, while new applications can access the data through the more flexible Web Services interface. As Web Services applications and services gain acceptance, and proprietary APIs are superseded by industry standard Web Services technologies, the data repositories become even more independent from the applications that access them. Consequently, the underlying DBMS functions, possibly even the DBMS itself, can change and evolve independently, without requiring changes to the applications accessing them. Maintenance costs are avoided improving the cost structure, and ultimately the competitiveness of the firm.

The benefit of providing accessibility to wider set of applications within the enterprise has an additional positive benefit. Data is often restricted to functional silos within the enterprise. For example, the service organization maintains one set of data regarding a particular customer, and the finance organization maintains data about the same customer, but in a separate DBMS. This data is often redundant and, in the worst case, inconsistent. Organizational turf wars and political issues aside, wrapping data repositories in Web Services technologies can extend data accessibility across these functional silos within the enterprise (Gottschalk et al., 2002). Consequently, a customer service representative can have a complete and consistent set of pertinent customer data in front of them when communicating with a customer, and both the speed and quality of customer engagements can improve. Providing a high quality customer experience is key to remaining competitive when customers can choose from many firms for a product or service.

Improved data accessibility via Web Services can also positively impact the ability of an enterprise to respond quickly to new business opportunities in the following way. As new combinations of data and functionality become available to Web Services applications, opportunities to add value in unique ways emerge. For example, imagine a firm specializing in publishing scientific educational content. The firm has two divisions, one that publishes content in the form of traditional perfect bound, paperback volumes. The other, to date has focused on publishing content, predominantly video and audio content, on CDROM. After each division has made their information assets available using Web service technologies, the opportunity arises to create a 3rd division. This division could access the information assets of the other two in order to create a web-based scientific learning center that combines text, audio and video assets available from the other two divisions. Taken one step further, the web-based learning center could allow subscribers to tailor their instruction to their own optimal leaning style by selecting the content that best suits them. In this example, a new application that consumes data made available through Web Services technologies (in particular SOAP) can focus on "what" data is available, rather than "how" to get it. Brand new APIs allowing access information assets in a new repository need not be built from the ground up. Providing customers options in when and where information is delivered can be an advantage for enterprises in highly competitive markets.

Similar opportunities are available within enterprises' IT infrastructures. For example, employee portals are becoming integral in disseminating information throughout the workforce of large enterprises. The information distributed often comes from a variety of organizations. An employee portal will have breaking news about the enterprise, links to benefits information, job opportunities, company directories and resources, and a host of other pertinent data for employees. Access to the information, if made available through Web Services interfaces can be accessed more easily, and the likelihood that the information is up to date and accurate is higher. In this sense employee portals are an intranet information aggregation application that can take advantage of service providers within the enterprise.

This phenomenon is the most powerful latent opportunity present in the adoption of Web Services. In *Out of the Box* Hagel (2002a) defines the n-squared problem as the exponentially increasing complexity of integration activity as the number of nodes to be integrated increases. We can apply this notion to the positive impact of making data more accessible by arguing that there is an n-squared opportunity, i.e., that the number of value propositions that can be constructed in response to customer needs increases as the number of nodes with unique data to contribute to the solution increases.

Improvements in data accessibility can have another, more subtle effect on customer focus. As more and more data becomes accessible to applications, redundancies will become more apparent. Emerging applications accessing data will tend to use the services that provide the highest quality and accurate data available. Over time, an evolutionary process of natural selection will take place where consumers gravitate toward the highest quality producers of data, in effect culling out the lower quality sources of data. For example, a Web service within the enterprise that consistently provides customers' street addresses truncated to sixteen characters will probably be rejected in favor of one providing customer street addresses of variable length. In addition, it need not take a long time for an information consumer to find and select the highest quality producer of a particular piece of data. Using UDDI and WSDL, multiple services can be located and compared, and the best match selected for integration.

6.1.1.3 Data transformation

Enterprises will, for varying reasons, continue to store data in on vendor specific platforms in native formats that are application specific. Wrapping data in Web Services technologies as described above is one of many transformation steps that may occur as data is processed in the context of individual applications (Gottschalk et al., 2002).

These transformation steps are made possible by the multitude of tools and technologies available for transforming XML data streams into other formats that are either more widely accepted or, formats that are application specific. The existence of DTDs and/or Schema for a particular XML data stream makes it possible to automate transformations from XML to any other desired format, using XSLT. In this way, applications can transform XML data into application specific formats without significant programming efforts.

These transformation steps are not new in the IT domain. Systems have always required that data be in a form that can be understood and processed in the context of the current application. What is new is that they can be automated using technologies like XSLT so that transformation engines do not need to be built from the ground up. Furthermore, while initial Web Services implementations will replicate data transformation processes within the context of a service, it is likely that with the growing sophistication of Web Services infrastructures, data transformation engines will reside on the network and offer transformation services for multiple applications and services across the enterprise (Forrester, 2002a). With the potential existence of data transformation engines in a SOA, not only is data liberated from the applications that use it, (Coyle, 2002) data is also liberated from the engines used to transform it.

Further, transformation from native DBMS format to XML has become easier as more vendors provide the ability to export XML data directly. DBMS vendors such as Oracle allow structured query language (SQL) commands that generate XML documents from relational data tables (Oracle, 2002). This type of data transformation is actually encapsulated in the off-the-shelf product, eliminating the need for a separate transformation engine. This trend is not limited to large suppliers of enterprise solutions. For example, MySQL, the database solution made available under the GNU General Public License (GPL) provides similar capability. The existence of transformation capability in DBMS reduces the need for enterprises to expend IT resources to develop transformation engines.

6.1.1.4 Data semantics

Today, most CIO's recognize that XML is not a complete solution to the problem of data exchange across the enterprise. It does, however, play a significant role in establishing shared meaning across the enterprise. XML tags can help in identifying information, but interpretation of what is between the tags still requires common understanding to be established (Hagel & Brown, 2001). For example, the XML tags <name> and </name> leave open to interpretation whether or not the content between them is first name, last name, or possibly both. Making the wrong assumption about the contents within the tags could potentially result in communications to a customer that begins with, "Dear William H. Jones", which, while not incorrect, is not as pleasing as "Dear William" or "Dear Mr. Jones" and may have an influence on the customer's perception.

Having said that, XML is more than just a new way to mark up data. The degree of independence of data in XML format is revolutionary. Data in XML is independent of rendering. It is independent of transport mechanisms. It is independent of programming languages and hardware/software platforms, and most important, it is independent of the applications that use it. It is simply a human readable, tag-based description of data. The tags are meant to serve a single purpose. Unlike tags in HTML, which combines display instructions with data tags, the tags in XML are used only to identify elements. And for all who can parse the tags, the data in an XML file is comprehensible. The key to realizing the full potential of XML-based Web Services on the Internet is the definition of meaningful, agreed upon vocabularies (Coyle, 2002).

Initially, the definition of XML vocabularies has focused on vertical industries such as finance, banking, science and technology. But the same opportunities exist within the enterprise. Just as the Open Financial Exchange (OFX, 2003) is meant to establish an interface with brokerage clients to download financial data into their local applications such as Quicken or TurboTax, an internally defined vocabulary can allow labor cost data, for example, to be exchanged between two organizations within an enterprise seeking to project costs for a project on which they will be collaborating. Vocabularies can be defined that accommodate a diverse set of users of data across the enterprise. And once a shared vocabulary has been established that contains the appropriate data, at the appropriate level of detail, multiple applications can store and process data within the context of their local application and exchange data with others with minimal transformation.

Let's say that multiple organizations within an enterprise have established a definition for a data element called "customer" that meets their local application requirements. The customer service organization, left to their own devices has requirements that would drive them to define "customer" in the following way:

```
<customer>
        <name></name>
        <address></address>
        <phone></phone>
        <email></email>
</customer>
```

A second organization, accounting, within the same enterprise may require both the company name as well as a contact name for this customer so that the customers' company name can be placed on invoices. This organization would describe the customer data element in this way:

```
<customer>
        <company_name></company_name>
        <contact_name></contact_name>
        <address></address>
        <phone></phone>
        <email></email>
</customer>
```

A third organization, sales, may require an additional level of detail in the contact name. First and last name may be useful for this organization so that they can personalize their communications to individuals. This organization would define the customer data in the following way:

```
<customer>
        <first_name></first_name>
        <last_name></last_name>
        <address></address>
        <phone></phone>
        <email></email>
</customer>
```

Given these three definitions of a customer data element, it is possible to define a vocabulary that contains a single customer data element that accommodates all three applications. The definition of the customer data element that would accommodate all three organizations' requirements would look like this:

```
<customer>
        <company_name></company_name>
        <first_name></first_name>
        <last_name></last_name>
        <address></address>
        <phone></phone>
        <email></email>
</customer>
```

The point of this illustration is that in order to realize the full potential of using Web Services within the enterprise, agreed upon vocabularies must be defined just as they must be defined for specific applications across enterprises just as is being done with vertical and horizontal standards. Once these vocabularies exist the incentive to create independent data repositories with data elements tailored to a specific set of requirements is reduced. The primary objective in the example above is to establish consensus on the answer to the question "what is a customer?" that satisfies the needs of all of the stakeholders within the enterprise. Enterprises can become more efficient, and ultimately more competitive, by eliminating misinterpretation of the meaning of data.

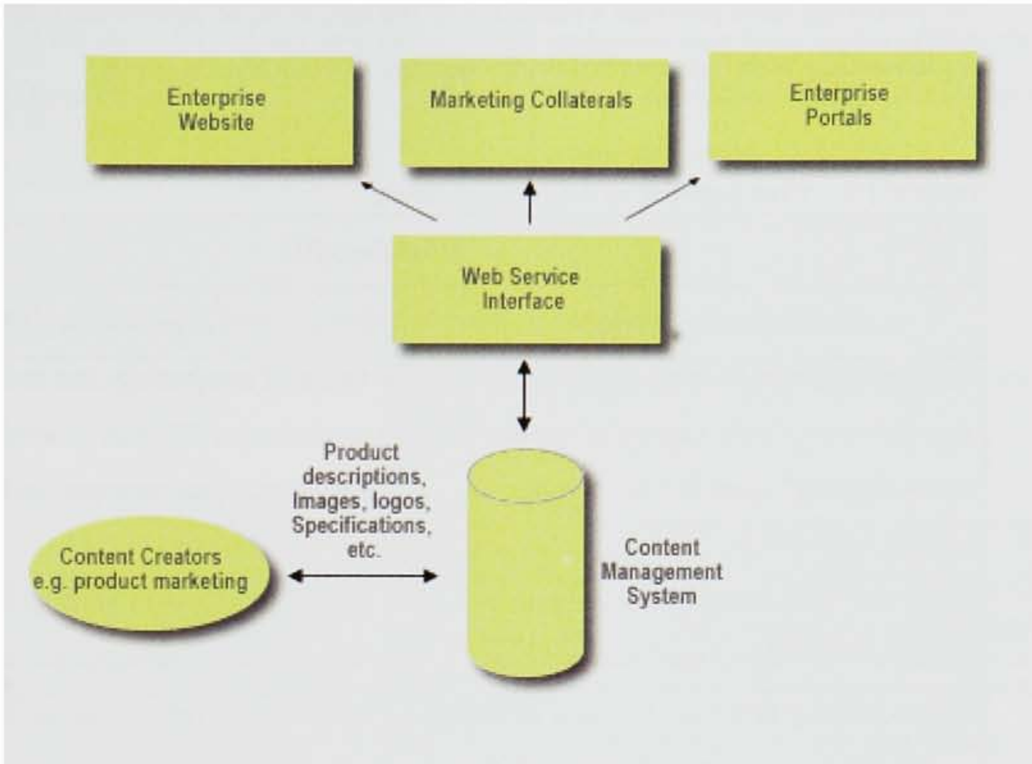
As shared meanings are established, the numbers of separate implementations of data repositories can be reduced, as well as the task of maintaining the data. In the past, data elements would have been replicated with varying levels of detail, in repositories on different platforms. One organization would store customer data elements in a network-accessible Oracle DBMS, another in an Access database on a local server. With the convergence of data repositories and data element definitions through standardized vocabularies and XML, redundant maintenance tasks can be eliminated. Avoiding maintenance costs contributes to the competitiveness of the enterprise.

6.1.1.5 Content management

Content management is a specialized case of data management that has significant implications to enterprise-wide efficiency and operation. Enterprises have digital assets that are often used for multiple purposes. For example, a product or service description may be rendered in a brochure, on the Web and on product packaging. Rather than developing and maintaining three separate assets based on their destination, a single digital asset can be developed and delivered to any application requiring that content. Enterprise-wide content management is increasingly important as companies find more and more channels for delivering product information. XML and Web Services can play a significant role in improving content management effectiveness within the enterprise.

XML data may be plain text, formatted documents, messages, images, numeric data, or any one of the wide varieties of digital representations for data we have devised to date, or will devise in the future. For example, product images, descriptions and specifications are rendered in marketing collaterals, on the company's website, and in catalogs. Content management systems based on XML allow enterprises to use a single source for content that can be "re-purposed" depending on the destination. Product descriptions can be tagged in XML format for flexible access, maintained in a single location, and delivered to requesting applications through Web Services interfaces as shown below.

Figure 6.4 - Content management systems



Content management systems based on XML and Web Services allow enterprises to manage digital assets in a way that not only improves accessibility and eliminates redundancies but also reduces the likelihood of inconsistent messages both inside and outside of the enterprise. Content assets developed independent of target destinations and propagated so that exact same message is delivered no matter what the context. This approach enables more effective version control, eliminates inconsistencies, reinforces branding and marketing messages, and enables a more efficient deployment of digital assets.

The unique attributes of content management are well suited to Web Services and a unified approach to integration within the enterprise. Digital content is a significant asset, and for some enterprises, such as Walt Disney, Universal Studios, AOL/Time Warner, and others, it can be a significant source of revenue when made available outside the enterprise.

6.1.2 Benefits summary

The positive impact of Web Services on data management is manifested in the form of cost benefits as well as added value in the form of new revenue opportunities. The following table provides a summary of the cost benefits associated with implementing Web Services within the domain of Data Management.

Table 6.1.1 – Cost benefits to data management

Cost Benefits	
Identified Benefit	*Explanation of Benefit*
Legacy data repositories can be made accessible to a wide variety of applications by "wrapping" them in Web Services technology.	Costs of integrating data sources through proprietary APIs can be avoided when data repositories are wrapped in Web Services.
Improved data accessibility can enhance the speed and quality of customer service.	Speeding access to pertinent data can reduce the cost of delivering products and services.
An underlying process of natural selection will make the producer of the highest quality data a preferred source.	Maintaining fewer sources with similar data is cost effective, reducing maintenance costs especially.
Transformation of data into application specific formats can occur automatically using readily available industry standard tools.	Costs of implementing proprietary data transformation engines can be avoided in some cases.
By delivering XML documents based on SQL queries, vendors are providing transformation capabilities.	Costs of implementing proprietary data transformation engines can be avoided in some cases.
Converging on common vocabularies and eliminating redundant repository implementations can reduce maintenance.	Costs associated with maintenance of multiple instances of data can be avoided.

Cost Benefits	
Identified Benefit	*Explanation of Benefit*
XML-based data repositories in a SOA provide the basis for enterprise-wide content management.	Treating digital content as corporate assets and managing those assets efficiently can reduce redundancies and maintenance and associated costs.

The following table describes the added value in the form of revenue opportunities of implementing Web Services within the domain of Data Management.

Table 6.1.2 – Business value to data management

Business Value	
Identified Added Value	*Explanation of Added Value*
Improved data accessibility can enhance the speed and quality of customer service.	Rapid and accurate response to customer needs can improve customer satisfaction and decrease the likelihood that they will defect to a competitor.
An underlying process of natural selection will make the producer of the highest quality data a preferred source.	Accuracy of data is a contributor to customer retention and repeat business.
Establishing shared vocabularies can improve the quality of customer engagements.	Personalized, one to one communication with customers reinforces the impression that individual customers are valued, and their repeat business is more likely.
XML-based content management repositories in a SOA can eliminate inconsistent messages.	Consistent marketing messages eliminate confusion and establish clear expectations for customers. Clear expectations are more easily met resulting in higher probability that the customer is satisfied.

Business Value	
Identified Added Value	*Explanation of Added Value*
Combinations of data heretofore not available can be processed in order to create new and unique customer value propositions.	Revenue generating opportunities are available to those who can create unique value propositions for customers. The basis of these value propositions is often not the "invention" or "discovery" of a new piece of information, but rather a combination of data that heretofore was not easily accessible. For example, time stamps, GPS data, and digital images have been accessible for decades. Combing these pieces of information in various ways provides unique concepts such as being able to verify that a photo was taken with a digital camera in a specific location, at a specific time. This combination of information in the hands of an insurance company adjuster, for example, adds value to the claims processing application.

6.2 SYSTEM INTEGRATION

As Web Services technologies mature and enterprises recognize the value in combining formerly stand-alone applications in the context of a SOA, system integration efficiency becomes more critical to application development and deployment. Enterprises must deal with multiple generations of heterogeneous platforms and applications. Without the existence of standards-based middleware such as Web Services to connect disparate systems, flexibility to integrate capabilities quickly to meet emerging business needs is reduced.

With the arrival of Web Services and SOA, development initiatives can in fact be composed almost entirely of the integration of available services in a unique orientation to create a new application. Some of the services to be integrated are new capabilities built using Web Services technologies, but many services will be standalone legacy applications. A desire to combine these separate applications creates the Enterprise Application Integration (EAI) challenge for IT organizations today (webMethods, 2003).

Enterprises recognize the relationship between relationship between Web Services and EAI. In a March 2002 study on Web Services awareness conducted by IDC 13% of all respondents said that the primary reason that their organization is adopting Web Services is to "solve internal application integration issues". An additional 17% of all respondents indicated that "solving integration issues with external organizations (partners or customers)" was the primary reason for adopting Web Services technologies (IDC, 2003a).

Improving the efficiency of system integration presents a significant opportunity for enterprises to increase competitiveness. In *Out of the Box* Hagel (2002) presents two challenges standing in the way of rapid integration required to quickly respond to business opportunities in the Internet age. The first challenge, the n-square problem, suggests that the complexity of integration activity increases exponentially, not linearly, as the number of nodes to be integrated increases. Therefore, as IT managers respond to rapidly growing needs to integrate sources of data and functionality to accommodate eCommerce and other Web-based applications, the time required to complete integration increases.

Hagel and Brown have suggested that grids of Web service functionality take the n-squared growth in complexity and "reduce it to 2n – all the end-points need master is how to connect its technology with the technology interface offered by the service". (Hagel & Brown,

2001) While not quantifiably proven this relationship of complexity to growth in complexity serves as the basis of significant opportunity in the deployment of Web Services technologies.

The second integration obstacle identified by Hagel is that the Web is evolving so rapidly that managers have a very difficult time anticipating the number and types of partners and connections that need to be established to respond to new business opportunities (Hagel, 2002a). For the IT managers, the challenge is the same. The number of potential connections and relationships are fewer inside the enterprise; however, the existence of well established silos of functionality and organizational responsibility impede the flexibility required to respond to new opportunities with novel solutions.

This section will describe the inherent characteristics of Web Services technologies that help to overcome these integration obstacles and make enterprises more competitive.

For the purpose of this analysis, system integration is broken down into the following sub-categories:

- **Service-Oriented Architectures (SOA)**
- **Platform independence**
- **Web Services and distributed computing**
- **Just-in-time integration**
- **Vendor-supplied web services frameworks**
- **Web Services management**
- **Legacy systems**

6.2.1 Technical analysis

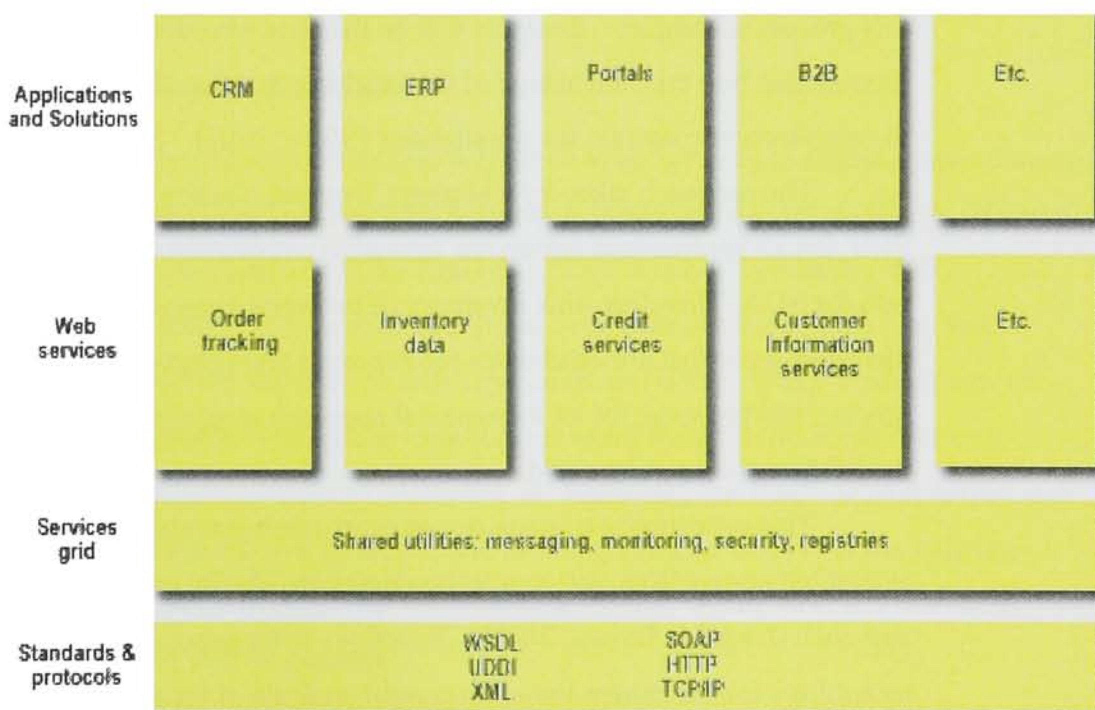
6.2.1.1 Service-Oriented Architectures (SOA)

Burbeck (2002) argues that the full potential of services, specifically Web Services, cannot be realized without being implemented in the context of a SOA. Further, he suggests that the efficient organization of components in a services architecture can only occur when they share certain organizing principles. And when such principles allow flexible, automated and dynamic methods of description, discovery and use, they serve as the basis of a SOA.

Furthermore, Burbeck equates a collection of collaborating services to a "theater production", where, after the script is chosen, actors are called upon to play their roles. The script is a definition of the business rules applied to solve a particular problem. The actors are the services that are either defined upfront, or "cast" at runtime based on a set of business rules. The SOA defines the organizing principles that determine what types of services play what roles at runtime.

One way to describe a SOA and the actors therein is to describe the components in terms of a "services grid". Figure 4.5 shows how a SOA is built on industry standards that allow the establishment of a grid consisting of a broad range of services that enable users and providers to form connections – regardless of hardware and software platforms - without appreciably increasing complexity as in the n-squared scenario described above (Hagel & Brown, 2002b). This SOA orientation and the resulting service grid provide a framework within which Web Services can be integrated. The complexity of doing so is lessened by the layers of standard protocols, shared utilities and capabilities exposed as services, all of which can be used as building blocks in the development of applications and solutions.

Figure 6.5 - Services grid



Of course, the benefits resulting from the establishment of a services grid as depicted above, are not realized until the architecture is largely in place. There is a period of time over which a transition occurs from the existing architecture in an enterprise to a SOA. These transitions do not occur quickly, and therefore a transition strategy must include a prioritized approach to identifying and implementing components with the greatest potential for reuse within the services grid.

We have learned from nearly two decades of experience with object-oriented methodologies and technologies that encapsulation is not enough. Services, like objects, will only be useful when they are described with semantically meaningful definitions and organized in ways that allow dynamic discovery and usage (Burbeck, 2002). Then, a services grid that enables economic value creation is possible.

Among the early adopters of Web Services technologies is Eastman Chemical. They have worked to convert an existing IT infrastructure consisting of a variety of applications by taking inventory of the company's application servers, assessing the primary functions of each, and, where appropriate, exposing that functionality for use by other applications using a Web Services interface. Once this process is complete, the users will be the ones who determine the business process that best take advantage of the available functionality, rather than having the architecture constrain the possibilities (Sliwa, 2003).

The approach taken by Eastman Chemical is noteworthy for two reasons. First, not all applications within the enterprise necessarily need to be integrated into the SOA. Therefore, this inventory of potential services systematically identifies those that are candidates for exposing and integrating. Secondly, this process sets the stage for an incremental approach to integrating new services into the SOA and the services grid.

This opportunity to stage investments when establishing a SOA is available because Web Services technologies are ideally suited to an incremental approach (Hagel & Brown, 2002b). Functions to be exposed using Web Services technology can be chosen based on careful analysis of the risks, rewards and potential impacts to the rest of the IT infrastructure. Over time, tightly coupled applications, middleware, and application services can be transformed into a flexible portfolio of applications and services that share standards such as data descriptions and connection protocols. In this desired state depicted in Figure 4.5, applications are free to discover and invoke services freely, without costly programming and integration steps.

This transformation from tightly coupled architectures and components within the enterprise to a more loosely coupled set of capabilities has been referred to as disintegration (Moschella, 2003). This disintegration within the enterprise is a pre-cursor to mature SOA and will enable IT departments to adopt a service orientation and to accelerate Web Services adoption and integration.

Anastasopoulos, et al. (2003) suggest that a domain-analysis approach be used to identify requirements common to applications within a particular domain. Following this process an organization can identify the components and sub-components that provide "generic capabilities" and are candidates to replace multiple, redundant capabilities within the domain. This is a clear advantage over the most recent wave of enterprise-resource-planning (ERP) software applications that are large, monolithic, and tightly integrated and rely on unified databases (Britt, 2002).

6.2.1.2 Platform independence

The primary driver that will allow Web Services to enable more efficient integration is the fact that Web Services is based on industry standards. Without agreed upon standards as the basis of a SOA, it is highly unlikely that interoperability will be achieved. It has been demonstrated in the past that interoperability depends on industry standards (Barry, 2002). Without agreed upon standards vendors implement proprietary systems that work best on a specific platform or set of platforms. By managing the XML, SOAP, UDDI and WSDL standards via the W3C, and implementing services over the existing Internet infrastructure, improved interoperability, and more efficient system integration will be possible.

An industry consortium, www.ws-i.org, supports demonstrations of platform independent Web Services-based solutions. According to the WS-I website, "The Web Services Interoperability Organization is an open industry effort chartered to promote Web Services interoperability across platforms, applications, and programming languages" (WS-I, 2003). The resources that the WS-I provides for those developing Web Services include sets of Web Services specifications that work together (profiles) to support specific types of solutions, sample implementations, implementation guidelines, and tools to support implementation and test.

These resources are available to anyone who becomes a member of the WS-I. Utilizing these resources in the development of Web Services will increase the likelihood of interoperable Web Services within the enterprise and speed system integration tasks. Just as important, individual IT organizations will not need to spend scarce IT resources on the development of test-beds and reference implementations for use in system integration and testing.

6.2.1.3 Web Services and distributed computing

Centrally managed corporate data repositories, resulting from widespread implementation of mainframe applications, were difficult to adapt to the client-server computing model that emerged in the 1990s. The industry attempted to adopt remote procedure calls to access data (and functionality) over the network through the development and use of technologies such as DCOM™, Corba™ and Java™ RMI. These technologies, while still in use today were intended to be vendor and platform neutral. That objective was never fully achieved, and therefore these technologies never became widely accepted. In addition, they never worked well over TCP/IP, the accepted standard for Internet communications (Coyle, 2002).

The emergence of XML has had an impact beyond the data representation for storage and exchange. It has also resulted in a significant change in the way we approach distributed computing by the application of XML to protocols. As a result, we now have SOAP and XML-RPC that are based on XML standards and totally independent of programming languages, operating systems and transport mechanisms (Coyle, 2002).

By using SOAP to access data through well-defined, platform independent interfaces using TCP/IP and the platform independent base of HTTP servers on which the Web is based, Web Services can provide accessibility to data independent of the transport mechanisms that deliver it, and individual software vendors providing the implementations of those transport mechanisms. In effect, that data delivered via a Web Service has been liberated from the means for delivering it (Coyle, 2002). The result is a reduced likelihood that interfaces will have to change to accommodate changes in the underlying data repositories, and thus an increased efficiency with respect to future maintenance.

Similarly, using Web Services to access data through platform independent interfaces improves data accessibility. Data are not only liberated from transport mechanisms, but also from applications. Improving data accessibility means that customer facing applications have access to a broader set of data to be used to improve the quality of customer interactions. For example, a large financial services firm sees improved customer service as the primary benefit of implementing Web Services. Leveraging legacy data currently implemented on an IBM CICS-based system can be delivered through Web Services interfaces to applications that previously did not have access to this data. The additional information can be used to personalize the customer experience and present a richer set of information that customers can use in making financial investment decisions. For this customer, according to IDC, "financial payback is a secondary concern to improving service levels" (IDC, 2002).

6.2.1.4 Just-in-time integration

More than liberating data from transport mechanisms, Web Services architectures can liberate connections between service producers and consumers from design time constraints. By using WSDL as the mechanism for describing a Web service, the advertised capabilities provided can change at runtime. Consumers can discover how information is exchanged with a Web service in real-time and react accordingly. Furthermore, by searching registries using UDDI for services that fulfill a runtime requirement, applications can select services dynamically at runtime, providing a capability referred to as Just-in-Time (JIT) integration (ZapThink, 2002).

As an example of how Web Services can enable JIT integration, refer again to the customer data element used above.

```
<customer>
        <account_number></account_number>
        <first_name></first_name>
        <last_name></last_name>
        <address></address>
        <phone></phone>
        <email></email>
</customer>
```

The WSDL file for a service that delivers information about a particular customer advertises, that given the account number for a customer as an input parameter, it will return the customer data defined above. One of two integration scenarios may exist. Either the service may make available a newly tagged item pertaining to the customer before applications are prepared to process that data, or applications may have been upgraded, expecting a tagged data item to be included in the customer information at some future date. The additional expected data element may be the name of the sales rep that calls on this customer.

```
<customer>
        <account_number></account_number>
        <first_name></first_name>
        <last_name></last_name>
        <sales_rep></sales_rep>
        <address></address>
        <phone></phone>
        <email></email>
</customer>
```

In scenario 1, the service can return any number of tagged elements of customer data. Only those tags recognized by the applications invoking the service will have meaning for that application. Similarly, applications expecting data elements to be included will not attempt to process those elements until the service delivers the appropriate data enclosed in the correct tags. In neither scenario is harm done. When the provider delivers the sales_rep element and the consumer expects the sales_rep element, new features and processing can take place just-in-time.

The reason for this robust behavior is found in the nature of XML and tagged data. In tightly coupled, transaction-oriented systems, the existence or non-existence of data must be carefully managed because it is often recognized by relative position in the data stream. With XML and tagged data, the existence of additional data initiates no attempt to process an element because the tag is simply unrecognized. This results in reduced brittleness, or an increase in robustness for both the producer and consumer of the data (Barry, 2002).

Just-in-time integration will additionally be supported by UDDI registries maintained for use by Web Services within the enterprise. These registries will be separate from any registry maintained to allow those outside of the enterprise to access publicly exposed Web Services. With a centralized registry for advertising internal services, the number of locations where they can be discovered is minimized, thus increasing the speed at which integration can be achieved.

6.2.1.5 Vendor-supplied Web Services frameworks

The availability Web Services capability in off-the-shelf products will aid enterprises in meeting their integration challenges when developing a SOA. J2EE™ and .NET™ represent the most significant frameworks for defining a comprehensive Web Services architecture. Other vendors are following suit. Database vendors are supporting Web Services interfaces for their DBMS products. Applications from Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) vendors are providing Web Services interfaces to simplify EAI. IBM's WebSphere™ and BEA's WebLogic™are examples of platforms that package a variety of tools on which applications can be developed and deployed using Web Services interfaces.

6.2.1.6 Web Services management

Development of a SOA and the integration of disparate services is more complex than discovering, exposing and accessing services. Tools to manage the available services are just as important as the services themselves. These tools are analogous to the data center monitoring tools that IT organizations depend on today for ensuring 24x7 availability of system capabilities. Eastman Chemical created an engine that goes far beyond simple registration of services for applications to access. The engine also manages "the security model, the orchestration between Web Services, debugging and monitoring of the services, fail-over capabilities, caching and data transformation" (Sliwa, 2003). This capability serves to add a level of maturity and is increasingly available as off-the-shelf from software vendors. The emergence of these Web Services management tools further eases the integration tasks and allows a level of transactional integrity that has been present in more tightly integrated systems within the enterprise. This capability is crucial to establishing a SOA, reducing integration times and improving competitiveness of the enterprise.

6.2.1.7 Legacy systems

With the arrival of each new wave of computing technology - mainframes, client-server, object-oriented, and now, Web Services - dealing with legacy systems and data becomes more complex. The issues around integrating mainframe-based legacy systems with client-server architectures are not completely addressed even as we enter into the transformation toward SOA. What's more, investments in IT architecture are significant and should not necessarily be abandoned merely due to the arrival of the next significant technological advance.

Legacy software, whether it is mainframe applications or applications that run on nodes in a client server architecture, presents unique technical issues and opportunities as enterprises consider the use of Web Services in their IT infrastructures. Approaches to handling legacy applications in the past have ranged from complete re-write of existing code to conform to new technologies and architectures, to the creation of elaborate intermediary layers of middleware and glue code to connect disparate systems.

Web Services provide at least two approaches to integrating legacy systems into a service-oriented architecture within an enterprise. The first option is to implement access to and from legacy applications using direct SOAP connections. The second is to use the frameworks described above - .NET and J2EE – to create a more seamlessly integrated system. Regardless of the approach, integrating functions embodied in legacy systems is a key component of establishing a SOA within the enterprise.

In the context of Web Services and SOA, legacy systems and applications play a valuable role in constructing the portfolio of services upon which new applications can be constructed (Coyle, 2002). The resulting SOA will be a mix of old and the new capabilities all wrapped in Web Services technologies that facilitate efficient and dynamic connections between disparate systems (Gottschalk et al., 2002). How effectively legacy systems and data are integrated to form new solutions will determine in large part the success of Web Services and to what extent enterprises will become more competitive as a result.

Legacy systems can be leveraged, not only to preserve investments in systems, but also to deliver the same services in a shorter period of time. For example, systems that formerly operated in batch mode, may be restructured to take existing core functions, expose them using Web Services technology, and deliver the same capabilities in real-time. Customers, who previously could not see the results of a transaction because a batch process that runs overnight had not yet been initiated, can now see transactions as they are processed (Arsanjani, et al., 2003).

It was the existence of pervasive legacy systems that drove Merrill Lynch to create a tool called X4ML that allowed their mainframe programmers to build and run Web Services capability to access CICS-based legacy applications. Merrill Lynch has literally thousands of mainframe applications that need to be integrated to support their rapidly growing eBusiness models. Once the Web Services interface is in place for a legacy application it can be considered for inclusion in the expanding grid of services available for composing products and services for customers (Sliwa, 2003).

Increases in competitiveness for enterprises come not only from the development of breakthrough technology, but also in leveraging valuable core competencies. Web Services allow enterprises to take advantage of these core competencies in ways that were previously not possible.

6.2.2 Benefits summary

The positive impact of Web Services on System Integration is manifested in the form of cost benefits as well as added value in the form of new revenue opportunities. The following table provides a summary of the cost benefits associated with implementing Web Services within the domain of System Integration.

Table 6.2.1 – Cost benefits to system integration

Cost Benefits	
Identified Benefit	*Explanation of Benefit*
Web Services architectures reduce the complexity of systems integration tasks.	Reduced complexity generally results in a reduction in the time and resources required to complete integration cycles.
Web Services technologies lend themselves well to an incremental integration of new capability.	Incremental integration reduces risk and costly integration phase overruns common in large-scale integration efforts.
Basing Web Services on industry standard technologies will enable more efficient system integration.	Industry standards increase the likelihood that interoperability can be achieved without developing custom code. Also, developing systems based on industry standards help to preserve technology and infrastructure investments.
Tools enabling developers to demonstrate and test interoperability are available from independent, 3rd party sources.	Costs associated with developing custom tools for integration testing can be avoided by purchasing 3rd party tools. (The make-buy decision process is the final determinant as to specifically how much can be saved.)

Cost Benefits	
Identified Benefit	*Explanation of Benefit*
Web Services liberate data from the underlying transport mechanisms, thus resulting in increased efficiency in the form of reduced maintenance.	Maintenance costs are a large portion of most IT department budgets. Decoupling data from data transport mechanisms means that data (and by implication business processes) remain constant as transport mechanisms change, and vice versa. Associated maintenance costs can be reduced.
By using WSDL and UDDI, Web Services can enable Just-in-Time integration.	Anticipating connections that may be required to support evolving business processes is an imprecise process. Connections implemented that are never included in business processes are a cost that can be avoided. Connections not anticipated may be implemented in a rush resulting in poor quality and incomplete functionality. Just-in-time integration using Web Services eliminates costs associated with the imperfect process of speculation.
Those seeking services will have fewer potential sources to examine.	The time required to identify the appropriate service to be integrated into a new solution will be reduced as analysts have fewer sources to search for those services and as services are increasingly well documented using WSDL.

Cost Benefits	
Identified Benefit	*Explanation of Benefit*
3rd party products increasingly provide Web Services interfaces.	Integrating functionality from 3rd parties made available through Web Services can reduce the integration cost of introducing new capabilities from 3rd parties.
3rd party vendors increasingly provide Web Services management tools.	Costs associated with developing custom tools for managing SOA can be avoided by purchasing off-the-shelf 3rd party software. (The make-buy decision process is the final determinant as to specifically how much can be saved.)
Web Services improve data accessibility making it easier for applications to present a rich set of information through customer facing applications.	Improved customer satisfaction results in improved customer retention. The cost to acquire new customers exceeds the cost to retain existing customers.

The following table describes the added value in the form of revenue opportunities of implementing Web Services within the domain of System Integration.

Table 6.2.2 – Business value to system integration

Business Value	
Identified Added Value	*Explanation of Added Value*
Web Services architectures reduce the complexity of systems integration tasks.	To the extent that system integration is completed faster, first mover advantage is possible to capture a new market.

Business Value	
Identified Added Value	*Explanation of Added Value*
Basing Web Services on industry standard technologies will enable more efficient system integration.	Compliance with industry standards can be leveraged as a selling point for customers who want to preserve their own investments.
By using WSDL and UDDI, Web Services can enable Just-in-Time integration.	Just in time integration supports the notion of rapid response to emerging business opportunities enabling enterprises to capture first-mover advantage.
Web Services improve data accessibility making it easier for applications to present a rich set of information through customer facing applications.	Improved customer satisfaction results in improved customer retention. Improved customer retention results in repeat business that may have otherwise been forfeited as customers searched and found competitive products that better met their needs. In addition, access to a wider variety of data presents new marketing opportunities such as up-sell and cross-sell.

6.3 APPLICATION DEVELOPMENT & DEPLOYMENT

In a March 2002 study on Web Services awareness by IDC 17% of all respondents identified the primary reason that their organization is adopting Web Services is to "deploy applications faster" or to "deploy applications at less cost" (IDC, 2003a). An additional 7% of all respondents indicated that "freeing up IT resources" was the primary reason for adopting Web Services technologies. Application development and deployment has traditionally been slow and inefficient as new IT technology is introduced, consuming significant resources in the process. A key component of increased competitiveness is reducing the resources required to develop and deploy new applications to respond to customer needs.

For the purpose of this analysis, application development and deployment can be further broken down into the following sub-categories.

- **Application development**
- **Application deployment**
- **Use of "best in class" capabilities**
- **Reduced it resource requirements**
- **Development outsourcing**
- **Service aggregation within the enterprise**

6.3.1 Technical analysis

6.3.1.1 Application development

Development and deployment of a wide variety of applications within the enterprise can be positively impacted by the implementation of Web Services. Characterizing the positive impact of Web Services implementations, IDC found that on average,

> "Benefits [of implementing Web Services] projected over three years include a reduction in costs of $39.7 million on an investment of $1.8 million, 22% faster time to deployment of key new applications, and an increase of 47% in developer efficiency," (IDC, 2003b).

The customers studied by IDC considered improvements in time to market to be a key factor in choosing a Web Services implementation strategy. The ability to deliver systems and solutions quicker, with fewer IT resources is a primary motivation for implementing Web Services within the enterprise. The study also found that Web Services applications will have positive impacts to application development within the enterprise in business process automation, enterprise application integration, accessibility to key business functions, best of breed technology selection, and location and device independence (IDC, 2003b).

As Web Services and SOA become increasingly prevalent, a Service-Oriented Development of Applications (SODA) methodology will become more critical in orchestrating implementations across the enterprise. SODA is a methodology that oriented toward implementation of SOA. Web Services technologies are the tools for implementing SOA. Whether developers are integrating legacy systems by bundling them in Web Services technologies, or developing services in a green field project using available Web Service frameworks, a SODA methodology will optimize the benefits of a SOA.

Reuse is a critical component of SOA/SODA. For years, systems designers have sought to increase the rate of reuse on IT projects to reduce the time and resources required to deliver new solutions. The modular nature of SOA provides a number of advantages with respect to application development, most notably modularity. Modularity allows systems designers and architects to consider substituting one module for another as the capabilities of alternative solutions evolve. And once a clearly superior component has emerged, more systems will specify the use of that component in their systems, thus increasing the frequency of reuse within an enterprise.

A second architectural characteristic of SOA/ SODA that will positively impact applications development is the extent to which applications, or consumers of Web Services, are decoupled from producers of those services. Tight coupling between applications and services encourages developers to make assumptions about the system providing the service. With Web Services all that is known about interfaces is specified in the WSDL file (ZapThink, 2002). No other assumptions should be made. When assumptions are eliminated, maintenance associated with correcting wrong assumptions is reduced.

A byproduct of deploying SOA in an enterprise is that with fewer interdependencies between modules and less focus on maintaining connections, more attention can be directed toward creating unique combinations of services. This benefit is described more thoroughly in section 6.4, but with respect to applications development and deployment within the enterprise it means that internal operations and functions are implemented with fewer resources. Reducing the overall IT support costs can reduce the cost of goods and services delivered thus improving enterprise competitiveness.

The increased focus on business processes also means that the role of Business Analysts becomes more significant, and the role of individual technologies, less so. The result is a level of abstraction in the available services that allows non-technical individuals to construct new value propositions. By focusing on the solution to customers' problems, rather than the systems and technologies comprising the solution Business Analysts can be more efficient.

The development and deployment of applications in a SOA will further be supported by the support of Web Services interfaces by producers of commercial off-the-shelf (COTS) software. Vendors such as Oracle, SAP and Seibel are already providing industry standard interfaces to functionality formerly accessed only through proprietary APIs (Oracle, 2002). Integration with industry standard Web Services interfaces provided by COTS vendors provide reduce the time required to develop new applications since code to interface through proprietary APIs does not have to be written, (and re-written) and tested.

Many enterprises may be inclined to take carefully managed, incremental steps toward Web Services application deployment. Services available on the Internet will, over time, evolve organically. Capabilities packaged as Web Services will change rapidly and will not be governed by the same organizing principles that can be defined in the context of a SOA within the enterprise (Burbeck, 2002). Within the enterprise, however, IT organizations can take an incremental approach to implementing Web Services by taking one or two functions that are not business critical and creating a Web service interface for applications that are already using the service but using a traditional API. By identifying low risk candidates for Web Services implementation and moving methodically toward the business critical functions, enterprises can reduce the risk of catastrophic failures such as those experienced when deploying large monolithic enterprise-wide systems (Sliwa, 2003).

6.3.1.2 Application deployment

Components of IT architectures in the past have often used static Application Programming Interfaces (APIs) to expose functionality for other components to use. Applications are often tightly coupled with these APIs (and the services they represent) not just technically, but from a business process standpoint as well. Changes in these interfaces often require a synchronized update to both the component offering the service as well as the applications invoking the service through the API.

Since Web Services support dynamic discovery through the use of registries and the ability to parse WSDL files at runtime to obtain information about the current interface, upgrades to a component or its interface will not require a synchronized upgrade and deployment process (ZapThink, 2002). Rather, a more automated deployment of upgrades to services can be executed without disruption of ongoing business processes.

6.3.1.3 Use of best in class capabilities

The use of Web Services technologies and SOA can serve to enhance quality and performance by best-in-class capabilities, making them available for others to use where appropriate. Within an enterprise duplicate capabilities are often implemented within a particular application domain. Exposing capabilities in an internal marketplace of available services will result over time in the survival of those that provide the highest quality performance. These surviving services represent the highest quality talent and expertise for the service, and applications that require the service will naturally gravitate toward these providers (Hagel & Brown, 2001).

The survival of the highest quality services will secondarily serve to strengthen those providers in several ways. Organizations with specific talents will be expected to further strengthen the performance of their services to reach a broader set of users. They will also need to hone very specific skills to maintain their leadership position. These specialized service providers will have a keen interest in training and recruiting for a very focused set of skills.

One example of a service that would benefit from the best-in-class capabilities is security. Firewall management, intrusion detection and other security services that require deep expertise in the area of security and must be efficiently deployed to best protect an enterprise's intellectual assets.

Using Web Services to expose interfaces to best-in-class capabilities will positively affect enterprise competitiveness with respect to customer satisfaction. By using best-in-class capabilities the performance and quality of products and services delivered to end users is more likely to meet, and possibly exceed, their expectations. Culling services that perform poorly or in a mediocre way and replacing those with the highest quality services available will improve overall from the perspective of the customer.

6.3.1.4 Reduced IT resource requirements

The use of Web Services encourages modularity and specialization. What follows, as is argued above is the emergence of best-in-class capabilities available for integration into a wide variety of applications. Corollary to this phenomenon is the reduction in incentive to invest in development of mediocre services that duplicate widely available world-class capabilities. Rather than replicating services, enterprises will quickly recognize the cost advantages of developing, maintaining and enhancing specialized services in fewer places. IT resources that were once replicated developing multiple instances of application specific services across the enterprise can be reduced or redeployed to assist in other areas. Further, the amortization of the cost of developing specialized Web Services can be amortized more broadly and more rapidly across the enterprise (Hagel & Brown, 2001).

6.3.1.5 Development outsourcing

Further addressing the application development bottlenecks facing enterprises is that Web Services exhibit "plug-and-play" characteristics. This attribute, coupled with the fact that even in these recessionary times outsourcing in general remains popular, presents an opportunity for enterprises to take advantage of outsourcing (Hagel & Brown, 2002a). The benefits of outsourcing – reduced costs, reduced asset exposure, and better return on assets – can be realized through implementation of Web Services and SOA.

In much the same way it is easier to introduce new peripherals to your desktop system, Web Services technologies will facilitate outsourcing development of components to third parties. The general trend toward outsourcing has been prominent for nearly two decades and has been motivated by the need to reduce costs and to gain access to rapidly changing technologies and skills (Moschella, 2003). The modular nature of SOAs will help to isolate functionality to be implemented in separate components with well-defined interfaces and functional requirements. In addition, data descriptions and connection protocols will be standards-based, ensuring that applications will connect freely without costly API development (Hagel & Brown, 2001). These characteristics of Web Services technologies are well suited to realizing potential economies of scale and increased competitiveness related to strategic outsourcing.

Outsourcing may take the form of integrating externally available Web Services to accomplish tasks within the enterprise. Over time, as enterprises expose capabilities using Web Services interfaces, certain commodity-like capabilities will become available over the Internet. These services do not necessarily need to be implemented locally, but rather may be invoked through Web Services interfaces from best-in-class providers. While proprietary or business-critical functions need to be managed carefully, and may not be good candidates for outsourcing in this fashion, many non-critical functions are (Barry, 2002).

The outcome of a make vs. buy decision with respect to a particular function will depend in part on the business model adopted by service providers, but in many cases integrating a best-in-class Web service accessible function (for example, the Google search engine, weather information from the National Weather Service, equity information from stock exchanges and route/map information from MapQuest™) will result in lower development costs as well as ongoing maintenance costs.

6.3.1.6 Service aggregation within the enterprise

The deployment of Web Services within an enterprise will require more than simply exposing scores of services to be accessed by emerging applications. Domain experts within the enterprise will integrate collections of highly specialized services. These specialists will choose from world-class services available from inside the organization as well as outside. In addition, they will add a variety of utilities need to manage the specialized capabilities represented in this aggregate collection (Hagel & Brown, 2001).

The CIO for an enterprise, working closely with the financial organization, could have specific responsibility for aggregation of services around the needs of the finance community. A collection of services – valuation, options analysis, currency hedging algorithms, etc. – could be identified from inside and outside the enterprise and made available for use by the financial community throughout the organization. In addition to providing this aggregate collection of services, utilities such as monitoring, comparative assessment of available services, brokering, conflict resolution, and others, could be provided to monitor and ensure the quality of services and performance is maintained over time. Furthermore, those responsible for service aggregation within an enterprise can constantly be using their specialized knowledge to be scanning the horizon for and evaluating emerging Web Services that provide new or better ways of meeting the corporation's requirements in a particular domain (Hagel & Brown, 2001).

The importance of a well managed aggregation and deployment of services becomes more and more critical as applications based on Web Services move toward mission critical functions. It is, of course, a problem if an employee portal is unable to display the current stock price of the company on an internal web site. It is another problem all together if a factory shuts down because of a breakdown in the supply chain or payment processing stops due to the lack of robustness in a financial service deployed within the enterprise. The use of Web Services aggregation within the enterprise can serve to improve the robustness of the services grid deployed within the enterprise, and ultimately to improve the quality of service delivered to the end customer as well.

6.3.2 Benefits summary

The positive impact of Web Services on application development and deployment is manifested in the form of cost benefits as well as added value in the form of new revenue opportunities. The following table provides a summary of the cost benefits associated with implementing Web Services within the domain of application development and deployment.

Table 6.3.1 – Cost benefits to application development and deployment

Cost Benefits	
Identified Benefit	*Explanation of Benefit*
Implementation of Web Services leads to increased reuse of components.	Effective reuse of software components has been demonstrated to reduce IT development and maintenance costs.
Application rework due to incorrect assumptions is reduced.	Loose coupling between producers and consumers of services and well-defined shared vocabularies reduces the likelihood that incorrect assumptions result in testing and integration failures.

Cost Benefits	
Identified Benefit	*Explanation of Benefit*
Implementation of SOA allows a portion of the burden of systems definition to be shifted to the Business Analysts.	Business analysts – who are closer to the customer requirements - can share in the process of defining solutions, and therefore increase the likelihood that the system meets their requirements. Rework associated with falling short of customer expectations can be avoided.
Web Services interfaces to COTS software simplify application development.	Web Services technology and standards, when built into COTS products will simplify development and, in some cases, eliminate the need to develop middleware components for data transformation and/or access to a service through a vendor-specific API.
Deployment of Web Services can be approached incrementally to manage risks.	Effective risk management, and specifically, avoiding the "big bang" approach to systems development can avoid systems downtime and related costs.
Service upgrades can be deployed automatically without disrupting ongoing business processes.	The nature of Web Services technologies help to avoid disruption costs associated with upgrading services.

Cost Benefits	
Identified Benefit	*Explanation of Benefit*
Implementation of SOA allows a portion of the burden of systems definition to be shifted to the Business Analysts.	Business analysts – who are closer to the customer requirements - can share in the process of defining solutions, and therefore increase the likelihood that the system meets their requirements. Rework associated with falling short of customer expectations can be avoided.
Implementation of SOA leads to increased focus on business processes.	Increase the likelihood that the system meets customer requirements avoids rework associated with falling short of customer expectations.

The following table describes the added value in the form of revenue opportunities of implementing Web Services within the domain of System Integration.

Table 6.3.2 – Business value to application development and deployment

Business Value	
Identified Added Value	*Explanation of Added Value*
Implementation of SOA allows a portion of the burden of systems definition to be shifted to the Business Analysts.	Customer expectations can be met more quickly when requirements are well known and addressed up front, reducing time to market with a viable product offering.
Service upgrades can be deployed automatically without disrupting ongoing business processes.	Fewer disruptions in services can translate into higher customer satisfaction and increased customer retention.

Business Value	
Identified Added Value	*Explanation of Added Value*
Exposing services inside the enterprise will encourage the use of world-class capabilities.	World-class capabilities are more likely to attract (and keep) customers.
Web Services aggregation can ensure a more robust, higher quality of service delivered to end customers.	Aggregation of services can result in a more complete and higher quality set of functionality for customers in a particular market segment increasing the opportunity for market leadership.
Implementation of SOA allows a portion of the burden of systems definition to be shifted to the Business Analysts.	Customer expectations can be met more quickly when requirements are well known and addressed up front, reducing time to market with a new product offerings.
Implementation of SOA leads to increased focus on business processes.	Customer expectations can be met more quickly when requirements are well known and addressed up front, reducing time to market with a viable product offering.

6.4 RESPONSE TIME

The value of Web Services and SOA goes far beyond the simple description, discovery and use of services. The value is derived not from the technology but rather from the business possibilities that result. Enterprises can focus on assembling services and solutions from a technology agnostic viewpoint, worrying only about the orchestration that makes it possible. This change in the approach to developing IT infrastructures will ultimately enable large scale innovation in business, especially eBusiness (Masud, 2002).

Innovation in customer-focused business solutions is becoming more and more critical. In the past IT technology suppliers played a dominant role in shaping infrastructures within an enterprise. Customers now play a more influential role as their requirements become more sophisticated and as they become more demanding and expect rapid responses to their needs. SOA and Web Services technologies play well in to this evolving trend and can play a major role in decreasing response time and increasing competitiveness of an enterprise (Moschella, 2003).

In order to remain competitive, enterprises must continually reduce the duration between the time a new opportunity is recognized and the delivery of a product or solution. As mentioned in section 5, there is a period during which a market leader has a privileged pricing position. This is often referred to as a "first mover advantage". As the Internet affords new prospects for delivery of information-based products and services, opportunities to capitalize on this first-to-market advantage will emerge. Capturing revenues during an uncontested window of time will be important for companies not only to maintain parity with their competitors, but also to gain competitive advantage.

The ability to reduce response time to new opportunities is largely an ability to adapt and reconfigure existing assets and systems in a new configuration. The ability to adapt components of an IT infrastructure can be referred to as the "option value", or the measure of an organization's ability to respond to fleeting opportunities in a volatile environment (Sifonis & Flinn, 2001). In their article "The New Economics of IT", Sifonis and Flinn observe;

> "In today's economy the ability of business models to be adaptive and scale quickly and effectively when required increasingly means the difference between extraordinary success or failure. The highest value

> role of IT in such an environment shifts from being just a vehicle for cost reduction, or even an enabler of individual initiatives, to a platform that supports the maximum number of business options. Therefore, IT architectural and system flexibility is of paramount importance, and must be explicitly recognized in IT strategies and investment programs. It is also absolutely essential that business executives now explicitly factor in the option value of IT investments. This value can ultimately dwarf other elements of the value equation, in some cases by orders of magnitude."

While the other sections of this analysis tend to be more focused on the cost savings aspects of implementing Web Services, this section is oriented toward realizing new revenue-generating opportunities, especially those that would have otherwise been difficult to respond to without the advantages afforded by these new technologies.

For the purpose of this analysis time required to respond to new opportunities can be further broken down into the following sub-categories.

- **Reduced time-to-market in existing market segments**
- **New business models and emerging market opportunities**
- **Reduced barriers to entry**
- **Leveraging existing intellectual assets**
- **Integration of mergers and acquisitions**
- **Response to regulatory requirements**
- **Compliance with vertical and horizontal industry standards**
- **Focus on business processes**

6.4.1 Technical analysis

6.4.1.1 Background

In *Out of the Box* Hagel asserts that, "what makes Web Services technology so powerful is its distinctive ability to help managers operate more flexibly and collaborate more successfully with business partners" (Hagel, 2002a). While Hagel was speaking primarily of business partners outside of the enterprise, this notion can be adapted to refer to partnerships within an organization. Web Services technology can help managers share data and services on the Intranet in the process of responding to emerging customer needs.

Hagel goes on to suggest that in the process of connecting people to the Web, businesses find that increasingly applications and data need to be connected "behind the scenes". Using Web Services technologies to create flexible, cost-effective connections, within the enterprise will have a positive effect as managers take advantage of their flexibility to rapidly address these emerging customer needs and opportunities.

Time-consuming technical challenges arise as enterprises attempt to increase the robustness of services offered to customers over the Internet. The level of difficulty increases when a company's web site moves from and information sharing facility, to an integrated eBusiness engine. Suddenly, up to date and accurate pricing data, product descriptions, user guides and product images are more critical than ever. Data from various sources, in diverse formats must now be accessed, processed and presented in real-time.

Behind the scenes, the eCommerce system must also have rapid access to inventory levels, shipping modules, credit information, and customer account data in order to complete transactions. Without technologies that support the location, delivery and translation all of this data in real-time, time-consuming development and integration tasks are required.

Without standard ways of accessing data and services through existing interfaces, as is provided by Web Services, sources must be identified, and an interface to each designed, implemented, tested and integrated in order to establish a new connection. This sequence needs to be repeated for each source of data required – pricing, inventory, customer data, etc. – and consumes scarce development resources in IT organizations. Once the initial implementation has been completed, changes in the architecture, the interfaces, or the structure of the data will likely necessitate a maintenance activity to keep the connection operable. In this scenario, the establishment of interfaces between service producers and service consumers is time consuming and error prone.

Further, managers will be hard-pressed to anticipate all of the connections that may be required as enterprises seek to respond to new business opportunities (Hagel & Brown, 2001). This inability to anticipate required connections implies that development of new interfaces cannot be initiated until the opportunity is known, immediately placing the development of these new connections on the critical path of a project.

Web Services architectures respond to this challenge by leveraging a handful of architectural principles: simplicity, loose coupling, heterogeneity, and openness – upon which SOA are based.

6.4.1.2 Reduced time-to-market in existing market segments

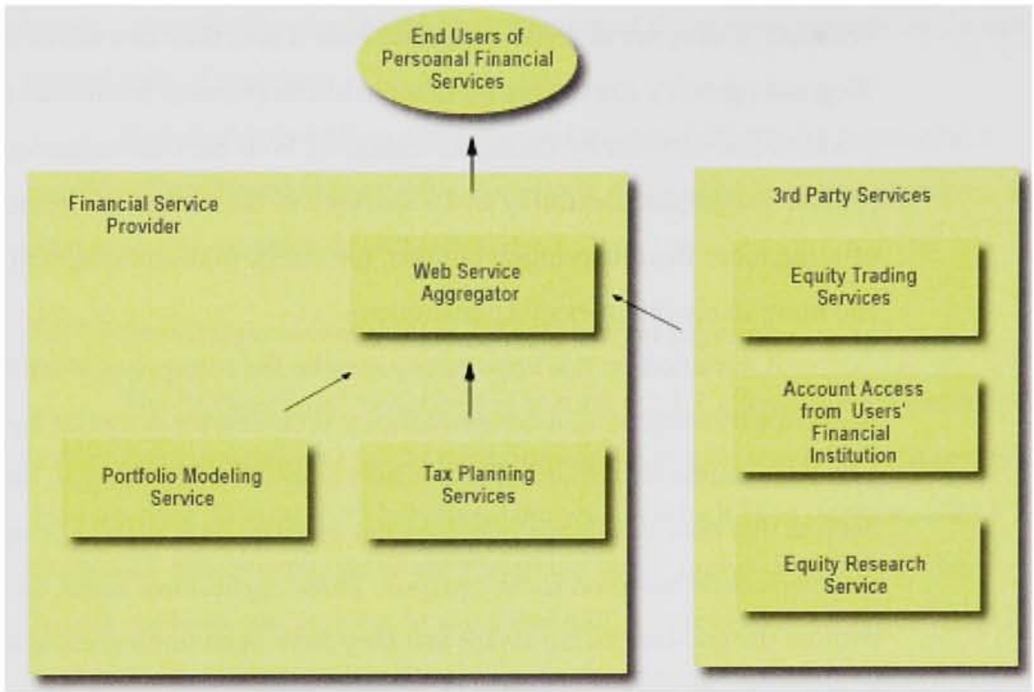
The "service grid" described in section 6.2.1.1 allows service producers and service consumers to form connections quicker than would have otherwise been possible without Web Services. One way in which Web Services afford enterprises new revenue generating business opportunities is to extend their current products and services in ways that had not previously been practical. The most obvious is for an enterprise to expose an existing capability using a Web Services interface. For example, Google has published a Web Service interface to allow others to access their data and search capabilities. Providing this service required only the publication of the interface to an existing capability and to ensure that the additional capacity required is available to service the new requests. Any enterprise with a unique computing capability, such as Google's proprietary search algorithms, has an opportunity to make that service commercially available by wrapping it in Web Services technology (ZapThink, 2002).

An additional way in which enterprises may take advantage of Web Services technologies is to extend the value proposition to their existing customers by adding Web Services capability to enhance an existing product or service. Rather than initiating a whole new value proposition, these Web Service-based capabilities can complement an existing product or service. One example of this class of opportunity may be for an online photo finishing company to allow consumers to print digital photos in a format of their choosing, to a convenient location. Printing services and capabilities can be advertised through a UDDI registry. Consumers can select locations at the time they decide to print one of their digital photos from a Web-based photo finishing company. If, for example, they are in Florida visiting grandparents, they can simply select the nearest Walmart. If they are at home in upstate New York, they can select the closest Wegmans grocery store. This service could complement traditional photofinishing products and services by taking advantage of Web Service technologies to provide geographic flexibility in the delivery of the customers' printed photos. By offering more flexible printing options, the online photofinisher will better serve, and more likely retain existing customers.

A variation on this opportunity may be for enterprises to extend a product or service by integrating a complementary Web Service provided by a 3rd party. The same online photofinisher above may consider integrating an imaging Web Service that provides image enhancement services such as red-eye removal or auto correction based on scene analysis. These capabilities could be implemented without the end-user being aware that they have been implemented by integration with a 3rd party service provider (ZapThink, 2002).

A simple example of the way in which an enterprise may take advantage of Web Services available from third parties is to look at a how a financial services firm may aggregate a variety of financial capabilities, both internal as well as external, to create a personal financial management portal. What the user may experience is shown in Figure 6.5 (Hagel, 2002b).

Figure 6.5 – Aggregation of Personal Financial Services



What is shown is that a financial service provider with core competencies in portfolio modeling and tax planning has utilized those capabilities in combination with the services available from third parties - equity trading, account access, and equity research – to create a personal financial service portal. The portal is essentially an aggregation of services available from both inside and outside of the enterprise.

The number of Web Services from 3rd parties that can be used to complement existing product offerings is rapidly increasing. www.Xmethods.com is a Web site that facilitates the development and deployment of Web services on the Internet. A full listing of available registered Web services will reveal hundreds of Web services currently available for integration into Web applications. The following table provides a sampling of categories of generic functionality available on the Web for integration into existing applications.

Table 6.4.1 - Web Services available from third parties

Generic Category	Specific Service	Available From
News and Weather	**Airport Weather** - information on weather conditions at airports around the world.	www.capescience.com
	IBM Weather Forecast Web services application - a Web service providing weather forecasts using data obtained from the National Weather Service.	http://www-106.ibm.com/developerworks/webservices/demos/weather/
	International News – Makes available articles from leading non-US media sources.	www.xignite.com
Investment Services	**Fixed Income Calculator** - Web service that calculates bond prices, yield to maturity, and other bond related functions.	http://www.financialwebservices.ltd.uk/
	Get Stock Market Headlines - Returns headlines for a specified list of US equities.	www.xignite.com
	Stock Quote Service – provides current quote for specified list of equities.	http://www.webservicex.net/

Generic Category	Specific Service	Available From
	Airfare Quote Service - Searches major airlines for the best available prices direct from their web sites.	http://www.dsdata.co.uk/
	Euro Conversion Service - Converts Euros to other currencies and back.	www.drbob42.com
Travel and Tourism	**Flight Information -** This SOAP/WSDL-enabled airfare server converts the data available from Yahoo's travel.yahoo.com site into XML, and adds a SOAP interface.	http://www.xmltoday.com/extras/flight/

In addition to those categories identified in the above table, services are also being made available for integration with Web applications for mathematical functions and calculations (area of a polygon, basic statistics, etc.), reference access (dictionaries, thesaurus, Bible, etc.), Web utilities (logging services, conversion utilities, security, etc.) and a wide variety of other generic functions. Careful selection and integration of these 3rd party capabilities can complement an existing product or service, quickly improving competitiveness of an enterprise's product or service.

6.4.1.3 New business models and emerging market opportunities

Of the architectural principles mentioned above – simplicity, loose coupling, heterogeneity, and openness - loose coupling is a primary driver in creating efficiencies that allow enterprises to respond quickly to new market opportunities. One of the key aspects of Web Services that enables the rapid establishment of these connections is that, while it is certainly possible to do so, the consumers of services need not "bind" to a service until runtime. Beyond static binding to services at design time, consumers may choose from at least two other approaches (Burbeck, 2002).

- Dynamic binding to a service: The application knows how to ask a registry/broker for the service, but details of the interaction such as the choice of transport protocol depend upon the service description returned by the broker at runtime.
- Dynamic choice of service: The application knows about the service to be used, but queries a registry/broker for a set of alternative service providers and the application chooses from the list at runtime.

This flexibility, while uncommon in IT infrastructures today, provides the basis of an architecture that will reduce response times to business opportunities. Connections between producers and consumers of services can be established quickly and independent of the functionality of either. This architectural characteristic is in contrast with enterprise architectures of the past that made the creation of new connections a time-consuming and expensive process (Hagel, 2002a).

6.4.1.4 Reduced barriers to entry

Reduced time to market, whether it be in existing market segments for emerging markets is really a bi-product of the fact that Web Services technologies reduce the barriers to entry into some information intensive, service oriented businesses. Whether or not reducing the barriers to entry in a particular market is a positive development clearly depends on an enterprise's current position within that market. If you are the leading provider of financial analysis tools available on the web, lowering the barriers to entry will increase competition and reduce your favored position. Conversely, if you own intellectual assets that are superior in quality, but have heretofore been inaccessible because of architectural characteristics within the enterprise, Web Services may afford you new opportunities to challenge market leaders without massive development costs being incurred.

Development costs are only one of the factors to be considered when considering entry into new markets. Capital equipment, maintenance, marketing and promotion, customer support and other costs must be considered as well. Web Services technologies however, can reduce the overall cost of entry making it possible for firms to compete in new markets.

6.4.1.5 Leveraging existing intellectual assets

Large enterprises that have extensive IT infrastructures, and decide to implement Web Services may have an opportunity not afforded small or medium-sized firms. As Web Services become more ubiquitous, access to raw computing resources (primarily computational and storage) where ever they may reside on the Internet becomes more feasible. An enterprise may find itself in a position to provide access via Web Services to mass storage capabilities, for example, that others can use as a repository rather than acquiring their own additional hardware. While this is not significantly different in theory from client-server distributed computing, the availability of an industry standard method of advertising and invoking these services is (ZapThink, 2002).

6.4.1.6 Integration of mergers and acquisitions

Integration of capabilities acquired through mergers and acquisitions has been a perpetual challenge for larger enterprises. Given the nature of Web Services technologies and architectures, it would seem evident that integration of acquired businesses and capabilities would be easier when both the acquiring and the acquired enterprise have established SOA architectures in implementing internal IT capabilities (Arsanjani, et al., 2003). This opportunity, of course, depends on SOA being much more pervasive in the future. Nevertheless, this possibility has significant implications for enterprises looking to quickly establish a competitive offering in a new market through acquisition.

Determining the value of a potential enterprise to be acquired is often a time consuming and complex process. Acquiring enterprises can better establish valuation of an acquisition by examining synergies, and evaluating ease of integration based on the extent to which capabilities are exposed using Web

Services technologies. An accurate assessment of this sort will not only allow enterprises to determine an accurate value of the organization to be acquired, it will also, as mentioned above, speed integration after the acquisition.

Similarly, for organizations that are to be acquired, easing integration requirements for acquiring enterprises by establishing a SOA may maximize the value (sale price) of the enterprise. In the event of an acquisition, both the buyer and the seller can build justification for a premium price based on the efficiencies gained by avoiding integration costs post acquisition.

6.4.1.7 Response to regulatory requirements

Regulatory compliance is another reason for organizations to base their IT infrastructures on SOAs. The Health Insurance Portability and Accountability Act (HIPAA) and the Graham-Leach-Bliley Act are examples of regulatory requirements where enterprises must respond in a timely manner or risk foregoing opportunities in the regulated areas. These requirements define a special class of emerging opportunity to which enterprises must respond.

The following table provides a sample of regulatory requirements and the industries for which they are applicable.

Table 6.4.2 - Regulatory requirements impacting IT infrastructure

Industry	Legislation
Banking and Investments	(Gramm-Leach-Bliley Act)
Insurance & Financial Services	(Health Insurance Portability and Accountability Act and Gramm-Leach-Bliley Act)
Healthcare	(Health Insurance Portability and Accountability Act)
Legal/Professional Services	(Health Insurance Portability and Accountability Act and Gramm-Leach-Bliley Act)

Industry	Legislation
Hi-Tech/Communications/Electronics	(Digital Millennium Copyright Act)
Retail/Consumer	(Health Insurance Portability and Accountability Act and Gramm-Leach-Bliley Act)
Discrete Manufacturing	(NIAP)
Public Sector	(Common Criteria/CCITSE-ISO 15408 and Family Educational Rights and Privacy Act)

Web Services technologies allow enterprises to establish a services framework ahead of time, without knowing what sectors may come under new regulatory requirements. For example, in the case of the Gramm-Leach-Bliley Act, there are new requirements for management of information and documents.

Some requirements dictated by such legislation require that certain information that currently resides within an enterprise (probably in electronic form) must be made available to consumers. Web Services technologies may reduce the time required to respond to such legislation by making the electronic information available through an interface that can be easily accessed and delivered to, say the corporate website for public consumption. Further, failing to comply with such requirements may result in fines and actions preventing an enterprise from offering particular products and/or services until such time that the requirement is satisfied.

The European Data Directive (EDD) developed by the European Parliament provides a framework for the collection, use and disclosure of personal data by enterprises conducting business in European countries (Harvey & Verska, 2001). Like Graham-Leach-Bliley, the EDD imposes strict requirements on the use of information. In fact, it goes beyond by stating, "Businesses may not transfer data outside the EU unless the recipient country provides adequate protection for the data". This is certainly a security issue, but more than that, it has significant implications for business processes. Enterprises that can demonstrate that their business processes conform to the requirements stated in the EDD will have opportunities to compete in the European region with new products and services while some of their competitors may not. Web Services can help be enabling the appropriate access to information to which the regulations apply.

6.4.1.8 Compliance with vertical and horizontal industry standards

Similarly, widely accepted standards, even those that are not specified by regulatory bodies, result in a more subtle impact of adopting Web Services and XML. As industry-wide standards emerge, potential customers, particularly in specific vertical markets, (health, education, finance, etc.) will insist that applicable existing XML vocabularies be used with the services they purchase. As enterprises adopt agreed upon data vocabularies for use within the enterprise, it will be advantageous to map those internal vocabularies to both vertical and horizontal industry standards. For example, a financial services firm would not want to define an XML vocabulary for exchanging securities data across organizations within the enterprise that is inconsistent with the OFX standard. [The Open Financial Exchange, OFX, is an XML-compliant specification for the exchange of financial data. The expectation is that it will be used by financial institutions as well as the developers of financial applications such as Intuit, Microsoft, and Quicken to standardize the exchange of data in the context of transactions such as bill payment, bill presentment, and investments tracking (OFX).] Rather, if data representation within the enterprise is compatible with existing industry standards, responding to new opportunities need not include the task of defining the data transformation from the internal to an external representation.

6.4.1.9 Focus on business processes

In the introductory comments it is stated that increased competitiveness is a matter of creating business processes, applications, and infrastructure that enable the efficient delivery of products and services. The bottom line with respect to SOA and Web Services is that the business process implications are just as weighty as the technical issues.

Modeling business processes and then combining those processes using service-oriented components is more complex than creating a portfolio of services and establishing the connections that allow service consumers to access service producers. Many business problems are solved by a series of transactions that must be choreographed from start to completion with the ability to interrupt, intercept and alter the sequence over (potentially) long periods of time (Udel, 2002).

More generally, W3C has recognized that "taking the next step in the development of Web services will require the ability to compose and describe the relationships between lower-level services. Although differing terminology is used in the industry, such as orchestration, collaboration, coordination, conversations, etc., the terms all share a common characteristic of describing linkages and usage patterns between Web services".

To that end, the Web Services Choreography Working Group published the first working draft of Web Services Choreography Requirements 1.0 on August 12, 2003. The Working Group is attempting to resolve the differences between work that has been done to date, including Microsoft's XLANG, used in BizTalk, IBM's WSFL (Web Services Flow Language), Sun, BEA, SAP, and Intalio's WSCI (Web Service Choreography Interface), Intalio's BPML (Business Process Markup Language), ebXML's BPSS (Business Process Schema Specification) (Udel, 2002).

The end result of this effort will be standard ways of coordinating specific types of transactions across a wide variety of services from a business perspective. Once this standard or set of standards has matured, enterprises will be able to quickly marshal resources (services) to appropriately address emerging business problems with very little concern for the underlying Web Services technologies.

One example of an industry's recognition of the need to coordinate business processes across entities is the Open Travel Alliance's (OTA) decision to adopt ebXML for conducting trading relationships, communicating data in common terms and defining and registering business processes (OTA).

According to the OTA "specifications that provide a message exchange between wholesalers and tour operators for booking holiday package tours… will utilize the ebXML secure messaging structure as a recommended reference envelope layer that provides OTA specification users with a unified, interoperable solution".

6.4.2 Benefits summary

The positive impact of Web Services on application response time is manifested in the form of added value with respect to new revenue opportunities. The following table provides a summary of the new revenue opportunities associated with implementing Web Services.

Table 6.4.3 – Business value to response time

Business Value	
Identified Added Value	*Explanation of Added Value*
Web Services architectures allow enterprises to expose existing capabilities quickly and easily to extend their products and services to a broader audience.	Extending existing value propositions can provide improve customer loyalty and increase customer retention.
Web Services architectures allow enterprises to complement their existing products and services.	Presenting new value propositions can provide new revenue streams from market segments adjacent to current target markets.
Web Services allows enterprises to complement their existing products and services with third party Web Service capability.	Using 3rd party capabilities to enhance existing product offerings creates more robust and competitive product offerings.

Business Value	
Identified Added Value	*Explanation of Added Value*
Web Services architectures allow the connections between data producers and data consumers to be established quickly and efficiently, reducing time-to-market with new products and solutions.	The first mover advantage affords a privileged pricing position, thus increasing revenues and profits.
Web Services allow IT-rich enterprises to make computing resources available to others.	Existing infrastructure capabilities may be the basis for new revenue streams for the enterprise.
Web Services can ease the burden of integrating capability acquired through acquisition.	Reduced integration costs can increase the value of a target acquisition lowering the cost of entry into new markets.
Acquiring enterprises can better value organizations by analyzing the extent to which the internal infrastructure has implemented Web Services technologies.	An enterprise acquiring a firm or portion of a firm with a well planned SOA and Web Services implementation can better value the individual revenue generating opportunities of the capabilities to be acquired.
Enterprises can maximize their market value through implementation of Web Services technologies.	An enterprise with a well planned SOA and Web Services implementation can place a premium on the market value in the case of acquisition. This premium increases overall shareholder value.

Business Value	
Identified Added Value	*Explanation of Added Value*
Regulatory requirements create a special class of opportunity to which enterprises, in many cases, must respond quickly.	Participating in certain markets, e.g. public sector, presents opportunities for profitable revenue. Without meeting certain prerequisites, participation in these markets is not available.
Internal data representations that map to industry standards reduce the time required to re-purpose data for internal use into data for exchange with external partners.	Adopting policies to represent internal data in industry standard formats effectively reduces the barrier to entry for a firm that decides to enter certain new markets. When considered in NPV calculations, payback period or other valuation techniques for evaluating business opportunities, this increases profitability.

7 Obstacles to web services implementation

7.1 Primary obstacles

Overcoming two key obstacles will increase the rate at which Web Services technologies are adopted. They are:

- Establishment of appropriate security standards and technologies to ensure the appropriate levels of access to and protection of information assets, and
- Orchestration of transactions and business processes

7.1.1 Web Services Security

The current set of Web Services technologies and frameworks does not adequately address basic security requirements. According to a Burton Group study (Schacter & Gable, 2002) Web Services falls short in the following areas: authentication (for both producers as well as consumers of Web Services), transactional integrity, non-repudiation, confidentiality and the ability to audit transactions. In the short term this will limit Web Services adoption for many enterprises to applications inside the firewall.

A number of security-related standards (XML-SIG, XKMS, Xenc, SAML, XACL, XrML) are in progress that will address the above issues as well as the use of public keys, digital signatures, access controls, encryption and Digital Rights Management (DRM) within the context of Web Services. Nevertheless, assembling secure applications and solutions from well-defined building blocks in a Lego-like fashion is still years away.

7.1.2 Orchestration of transactions and business processes

Another key aspect of Web Services that has yet to be sufficiently addressed is the orchestration of transactions, or workflow, across participants in a SOA (Udel, 2002). What business rules will be applied? How are solutions assembled from the available technical alternatives? How can potential solutions be modeled to understand pros and cons of potential approaches?

Similarly, the question of how the testing of Web Services across enterprises can be accomplished. None of the Web Services standards that currently exist describes the semantics of using a collection of services to support complex business processes. The use of Web Services is currently atomic and stateless in nature. Addressing the issue of

Web Services choreography will allow services to be combined into more complex solutions that better support business processes (Leyman & Roller, 2002).

7.2 OTHER OBSTACLES

Other obstacles to realizing the full benefit of Web Services remain. A report from the Delphi Group entitled "Web Services and Business Process Management" documents the responses of IT managers asked to identify the major challenges associated with implementing Web Services (Delphi Group, 2002). The top five obstacles they identified were:

- Inexperience developing service architectures
- Overcoming barriers in organizational culture
- The emergence of multiple standards for implementation
- The perception that the business case for Web Services has not been sufficiently established
- The difficulty in managing the required relationships with other organizations (both internal and external)

In addition to these obstacles, based on our research we would add the following.

- Development of meaningful shared vocabularies useful from both vertical and horizontal business process perspectives
- Alignment of business strategy with IT investments in the development of SOA
- Quantifying IT investment value associated with having a diverse set of options
- Web Services systems management

7.2.1 Inexperience developing service architectures

As with all new waves of technology, Web Services and SOA introduce a learning curve. Web Services is different, however, in that the learning curve is not simply a technical one. It involves multiple disciplines – product development, IT, business strategy, product planning, supply chain management, etc. - across the enterprise. It includes internal organizations as well as external. The broad implications impact nearly all areas of the enterprise making the transition to services more complex than other technological advances in the past.

7.2.2 Overcoming barriers in organizational culture

Change within the enterprise is often disruptive and is often met with resistance. Web Services and SOA represent a very different way of organizing and delivering capabilities and while we have some albeit limited experience with Web Services, bona fide successes are few. To counteract resistance to change the value of initial Web Services implementations must be assessed and publicized. Until such time that an empirical body of data supports the ability of Web Services to reduce inefficiencies and provide new revenue opportunities, resistance will remain.

7.2.3 Multiple implementation standards

The existing standards for XML, UDDI, WSDL and SOAP leave sufficient freedom for individual implementations that would not be interoperable. Key stakeholders (Microsoft, IBM, Sun and others), in the context of industry collaborations (W3C, WS-I, and others) must focus on defining ways in which implementation specific incompatibilities can be addressed.

Hagle articulates this risk as the possible fragmentation of standards and protocols. The risk, he suggests, is that vendors will take divergent paths by seeking to introduce proprietary versions rather than supporting the broader proposed industry standards (Hagel, 2002a).

7.2.4 Insufficient business case for Web Services Implementation

The intent of this thesis is to provide a framework that can serve as the basis for a value assessment of Web Services investment decisions. While not complete, this framework and others like it can serve as the basis of a more methodical approach to establishing a business case for Web Services implementation.

7.2.5 Managing relationships with other organizations

The Delphi Group report presumes that Hagel's definition of the n-squared problem (the exponentially increasing complexity of integration activity as the number of nodes to be integrated increases) (Hagel, 2002a) can also be applied to integrating organizations and corresponding business processes. Successful Web Services implementations and SOAs will require a significant level of collaboration across stakeholders in order to realize the full potential of the technology.

7.2.6 Development of meaningful shared vocabularies

Not everyone agrees that XML, the very core of Web Services, has not yet gained the widespread acceptance that is required for the touted benefits of Web Services to be realized. XML is a way of describing languages; a meta-markup language. Interoperability between service providers and service consumers is highly dependent on the vocabularies that can be described using XML and at the core of those vocabularies are business issues – issues that remain complex no matter what technologies are applied. It does not necessarily hold true that because two enterprises (or organizations within an enterprise) would agree on an XML Schema or DTD for a particular application, that those two can successfully persuade others to adopt the same standard (Young & Shirky, 2002). Without widespread agreement, entire industries run the risk of the variants. As a result, interoperability may suffer.

Hagle argues that lack of adoption of common vocabularies would result in a "balkanization" similar to the experience with EDI. Shared vocabularies must overcome the competitive dynamics present in the marketplace and a high degree of trust must be established before such agreements can be reached (Hagel, 2002a).

7.2.7 Alignment of business strategy with IT investments

In their IBM Systems Journal article, Luftman, et al. (1993), argue "business success depends on the linkage of business strategy, information technology strategy, organizational infrastructure and processes, and I/T infrastructure and processes". With respect to Web Services, the implication is that the broad, positive impact that many people expect will only happen if and when IT is considered as a key component in the delivery of value to customers, rather than a support function.

7.2.8 Quantifying IT investment value associated with having options

While articulating the value associated with having available options may not be a significant barrier to the advancement of Web Services and SOA, a methodical approach to assigning value to those options may represent a catalyst that would speed adoption cycle. Real options valuation represents an opportunity to turn a heretofore qualitative assessment of IT technology investments into a quantitative one (Copeland & Antikarov, 2001).

8 Conclusions

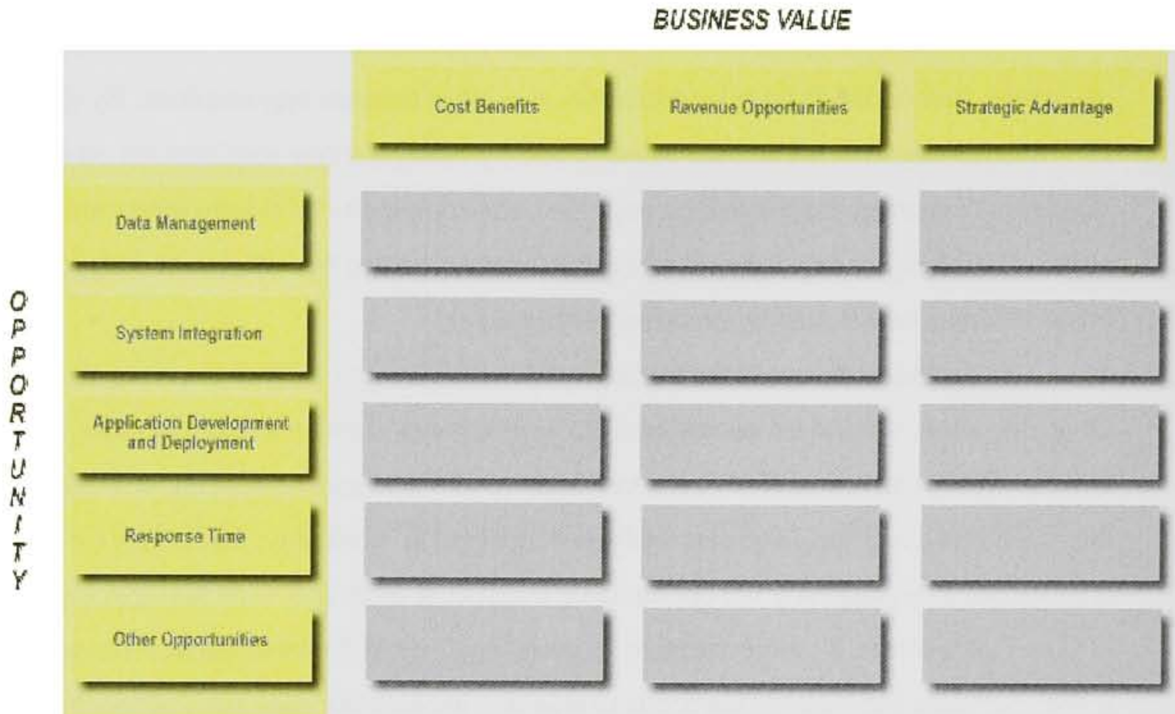
This thesis has shown that Web Services implementations will contribute value to the enterprise, both in the form of cost benefits as well as revenue opportunities. By implementing IT infrastructures that are service-oriented and by basing service architectures on industry standards data management will be improved, the complexity of system integration will be reduced, application development and deployment costs can be reduced, and responses to new business opportunities can be delivered more quickly.

To further understand the business related advantages of Web Services, the benefits identified above should be quantified in an appropriate valuation model (Samtani & Sadhwani, 2002). Net Present Value (NPV) is a valuation model that may be applied; however, other approaches, such as Real Options analysis (Copeland & Antikarov, 2001) may be more appropriate in highlighting the option value inherent in SOA and Web Services implementations.

Furthermore, a comprehensive financial analysis of the business benefits associated with a Web Services implementation needs to take into account not only the business benefits (both tangible and intangible) and costs incurred, but also must systematically account for assumptions, sensitivities, risks, and contingencies.

In addition, further work may establish a more complete value framework within which Web Services technologies may be evaluated. One example of the way that the value framework used in this thesis could be extended is to use the following as proposed in section 2.

Figure 8.1 – Extended Value Framework



By adding the *strategic advantage* component of business value, investments that yield neither cost benefits nor immediate revenue opportunities can be evaluated for their strategic potential. One example may be wrapping a capability with Web Services technology even though there is no immediate plan to expose that service to other applications. This adds to the portfolio of available components that can be utilized to respond to future, unanticipated business opportunities.

In addition, there are other areas of opportunity for which an IT investment such as Web Services can be assessed. One example may be response to global opportunities. Certain IT investments could impact an organization's ability to easily respond in global markets creating efficiencies in the areas of translation and localization. This could drive both cost advantages as well as new revenue opportunities in the global marketplace.

A second example of an area of opportunity may be the impact of an investment on system and process automation. IT investments often impact the extent to which human intervention is required to accomplish the delivery of a product or service. Establishing automated linkages and connections between systems and processes can yield cost benefits by reducing labor costs.

The complete framework appropriate to a value-based assessment will vary from enterprise to enterprise. Establishing a framework, however, to identify the key components of business value as well as the areas of opportunity is an essential first step in completing an assessment of the value of each investment.

Appendix A - Acronyms

API - Application Programming Interface

COM - Component Object Model

CORBA - Common Object Request Broker Architecture

COTS – Commercial Off-the-Shelf

DBMS - Database Management Systems

DCF – Discounted Cash Flow

DCOM - Distributed Component Object Model

DRM – Digital Rights Management

DTD - Document Type Definition

EAI - Enterprise Application Integration

EDD - European Data Directive

ERP - Enterprise Resource Planning

GPL - General Public License

HIPA - Health Insurance Portability Act

HTML - Hypertext Markup Language

HTTP - Hypertext Transfer Protocol

IT - Information Technology

NPV – Net Present Value

OFX - Open Financial Exchange

OTA – Open Travel Alliance

RMI - Remote Method Invocation

RPC - Remote Procedure Call

SAML - Security Assertion Markup Language

SOA - Service Oriented Architecture

SOAP – Originally an acronym for Simple Object Access Protocol

SODA - Service Oriented Development of Applications

SQL - Structured Query Language

TCP/IP - Transmission Control Protocol/Internet Protocol

UDDI - Universal Description, Discovery and Integration

W3C - World Wide Web Consortium

WSDL - Web Services Description Language

WS-I - Web Services Interoperability

XACL - eXtensible Access Control Language (aka XACML - eXtensible Access Control Markup Language)

Xenc – XML encryption

XKMS - XML Key Management Specification

XML - Extensible Markup Language

XML-SIG - XML signature

XrML - eXtensible rights Markup Language

XSLT - Extensible Style Language Transformation

Appendix B - Resources

Apache Web Services project - http://ws.apache.org/

CBDIforum.com: Independent think-tank on business software creation, reuse and management - http://www.cbdiforum.com/index.php3

CoverPages: OASIS sponsored Markup Languages Resource - http://xml.coverpages.org/

DeveloperWorks/IBM: - http://www-106.ibm.com/developerworks/

Electronic Business using eXtensible Markup Language (ebXML) - http://www.ebxml.org/

Graham-Leach Bliley Act - http://www.senate.gov/~banking/conf/

IBM Institute for Business Value - http://www-1.ibm.com/services//bis/ibv.html

IBM DeveloperWorks/Web Services - http://www-106.ibm.com/developerworks/webservices/

IBM's Specification for Web Services Security (WS-Security) - http://www-106.ibm.com/developerworks/webservices/library/ws-secure/

Impact Technologies: http://www.impact-tech.com/

Internet Security Alliance - http://www.isalliance.org/

ManyWorlds: Business Thought Leadership – http://www.manyworlds.com

Microsoft Web Services Developer Center - http://msdn.microsoft.com/webservices/

OASIS – eBusiness standards consortium http://www.oasis-open.org/home/index.php

OMG, The Object Management Group Consortium - http://www.omg.org/

Oracle's Web Services Technology Center - http://otn.oracle.com/tech/webservices/index.html

Open Financial Exchange - http://www.ofx.net/

Real-Options.org http://www.real-options.org/

SOAP/RPC.com - http://www.soaprpc.com/

UDDI Consortium – http://www.uddi.org

Web Services Journal - http://www.sys-con.com/webservices/

WebServices.org: A Web Services industry portal - http://www.webservices.org/

World Wide Web Consortium (W3C) – http://www.w3c.org

WS Index: Web Services and Semantic Web Resources - http://www.wsindex.org/

Web Services Architect – http://www.webservicesarchitect.com/

Web Services Interoperability (WS-I) Organization – http://www.ws-i.org

XMethods - www.xmethods.com

XML.com – http://www.xml.com

XML Trust Center - http://www.xmltrustcenter.org/index.htm

XML Web Services Security Forum - http://www.xwss.org/index.jsp

Bibliography

Applegate, L. M., McFarlan, W. F., McKenney, J. L. (1996). *Corporate Information Systems Management: The Issues Facing Senior Executives. 4th Edition*. Chicago, IL: McGraw-Hill Professional Publishing.

Applegate, L. M., McFarlan, W. F., Austin, R. D. (2002). *Creating Business Advantage in the Information Age, 4th Edition*. Chicago, IL: McGraw-Hill Professional Publishing.

Applegate, L. M., Bock, G. (1995). Managing Information Technology in the 1990s. Boston, MA: *Harvard Business School Publishing*.

Anastasopoulos, M., Bayer, J., Bunse, C., Girard, J-F., Isabel, J., Muthig, D., Sody, P., Tolzmann, E. (October 2002.) Software Evolution in Practice: Adding Web Functionality to a Legacy System. Proceedings of the Net.ObjectDays (NODE'02), Erfurt, Germany, October 2002. *Fraunhofer Institute Experimental Software Engineering (IESE)*. IESE-Report No. 055.02/E, pp. 1 - 35.

Arsanjani, A., Hailpern, B., Martin, J., Tarr, P. (March 2003). Web Services: Promises and Compromises. *Queue Magazine (Association for Computing Machinery)*. **1**, (1), pp. 170–177.

Barry, D. (2002). *Web Services and Service-Oriented Architecture: The Savvy Manager's Guide*. St. Louis, MO; Morgan Kaufmann Publishers.

Boehm, B.. (March 2003). Value-Based Software Engineering. *ACM SIGSOFT Software Engineering Notes*. **28**, (2), pp. 1-11.

Boehm, B., Huang, L. G. (March 2003).Value-Based Software Engineering: A Case Study. *IEEE Computer Society* . **36**, (3), pp. 33 – 41.

Boehm, B., Sullivan, K. J. (2000). Software Economics: A Roadmap, Proceedings of the conference on the future of Software engineering. *(ICSE) 2000*. pp. 319 - 343.

Britt, F. F. (2002). *Multiplying Business Value: The Fusion of Business and Technology*. Retrieved August 15, 2003, from http://www-1.ibm.com/services/strategy/files/ibv_fusion_ud.pdf.

Burbeck, S., (2002). *The Tao of eBusiness Services: The Evolution of Web Applications Into Service-Oriented Components with Web Services*. Retrieved August 2003, from http://www-106.ibm.com/developerworks/webservices/library/ws-tao/.

Copeland, T., Antikarov, V. (2001). *Real Options: A Practitioner's Guide, First Edition*. New York, NY: Texere.

Coyle, F. P., (2002). *XML, Web Services, and the Data Revolution*. Boston, MA: Addison Wesley.

Delphi Group. (2002). *Web Services and Business Process Management*. Boston, MA.

Documentum. (2003). *Developing Web Services with Documentum*. Retrieved August 15, 2003 from, http://www.documentum.de/products/collateral/platform/wp_tech_web_svcs.pdf.

Forrester. (May, 2002a). *Web Services True Costs Revealed.* Cambridge, MA. Forrester Research, Inc.

Forrester. (May, 2002b). *Web Services Platform Shootout.* Cambridge, MA. Forrester Research, Inc.

Gartner Group. (March, 2003). *Web Services Key Issues for 2003.* Stamford, Connecticut. Gartner Inc.

Gottschalk, K., Graham, S., Kreger, H., Snell, J. (2002) Introduction to Web Services Architecture. *IBM Systems Journal*. **41**, (1), pp. 170–177.

Hagel III, J. (2002a). *Out of the Box*. Boston, MA: Harvard Business School Publishing.

Hagel III, J. (2002b). The Strategic Value of Web Services. *The McKinsey Quarterly.*

Hagel III, J., Brown, J. S. (October 2001). Your Next IT Strategy. *Harvard Business Review.*

Hagel III, J., Brown, J. S. (2002a). *Orchestrating Business Processes - Harnessing the Value of Web Services Technology*. Retrieved August 15, 2003, from http://www.manyworlds.com/.

Hagel III, J., Brown, J. S. (2002b). *Service Grids: The Missing Link in Web Services*. Retrieved August 15, 2003, from http://www.johnhagel.com/paper_servicegrid.pdf.

Harvey, J. A., Verska, K. A. (2001). *What the European Data Privacy Obligations Mean for U.S. Businesses*. Retrieved August 15, 2003, from http://www.gigalaw.com/articles/2001-all/harvey-2001-02-all.html.

IBM. (December, 2003). *Web services (SOAP) with VSE/ESA.* Retrieved December 1, 2003, from http://www-1.ibm.com/servers/eserver/zseries/os/vse/software/vsesoap.html.

IDC. (March, 2003a). *U.S. Web Services Market Analysis.* Framingham, MA: IDC.

IDC. (March, 2003b). *IBM and the Strategic Potential of Web Services.* Framingham, MA: IDC.

IDC. (March, 2002). *Web Services Adoption Timeline and Related Business Opportunities.* Framingham, MA: IDC.

Lehman, M.M., Belady, L. (1985). *Program Evolution: Processes of Software Change.* London: Academic Press.

Leymann, F., Roller, D. (2002). *Business Processes in a Web services World.* Retrieved December 11, 2003, from http://www-106.ibm.com/developerworks/webservices/library/ws-bpelwp/.

Linthicum, D. S. (1999) *Enterprise Application Integration, 1st Edition.* Boston, MA: Addison-Wesley.

Luftman, J. N., Lewis, P. R., Oldach, S. H. (1993). Transforming the Enterprise: The Alignment of Business and Information Technology Strategies. *IBM Systems Journal.* **32** (1), pp. 198-220.

Masud, S. (2002). *Web Services: An Elephant in the Dark.* Retrieved August 15, 2003, from http://searchwebservices.techtarget.com/originalContent/0,289142,sid26_gci861248,00.html.

Moneypenny, N. (October, 2003) *Understanding Value and How to Create it on a Sustained Basis.* Retrieved October 15, 2003, from http://www.manyworlds.com/.

Moschella, D. (2003). *Customer-Driven IT.* Boston, MA: Harvard Business School Press.

Newcomer, E., (2002). *Understanding Web Services: XML, WSDL, SOAP, and UDDI.* Boston, MA: Addison Wesley.

Open Financial Exchange (OFX). (2003). *About OFX.* Retrieved August 15, 2003, from http://www.ofx.net/ofx/ab_main.asp.

Open Travel Alliance (OTA). (2001). *Open Travel Alliance Endorses ebXML.* Retrieved August 15, 2003, from http://www.ebxml.org/news/pr_20010801.htm.

Oracle. (2002) *Oracle9i Database Summary.* Retrieved August 15, 2003, from http://otn.oracle.com/products/oracle9i/pdf/Oracle9i_Database_summary.pdf.

Orth, G. (2002). *The Web Services Framework: A Survey of WSDL, SOAP, and UDDI.* Unpublished master's thesis, Vienna University of Technology, Vienna, Austria.

Peng, C. S., Chen, S. K., Chung, J. Y., Roy-Chowdhury, A., Srinivasan, V. (1998) Accessing Existing Business Data from the World Wide Web. *IBM Systems Journal.* **37** (1).

Samtani, G., Sadhwani, D. (August 2002). *Web Services Return on Investment.* Retrieved August 15, 2003, from http://www.webservicesarchitect.com/content/articles/samtani07.asp.

Sawhney, M. (July 2003). *Fundamentals of Value.* Retrieved August 15, 2003, from http://www.cio.com/archive/070103/gain.html.

Schacter, P., Gebel, G. (2002). *Web Services Security.* Midvale, UT. The Burton Group, Inc.

Seacord, R., Plakosh, D., Lewis, G. (2003). *Modernizing Legacy Systems.* Boston, MA: Addison Wesley.

Shapiro, C., Varian, H. (1999). *Information Rules.* Boston, MA: Harvard Business School Press.

Sifonis, J., Flinn, S. (2001). *The New Economics of IT: Option Value is the Value.* Retrieved August 15, 2003, from http://www.line56.com/articles/default.asp?NewsID=3192.

Sliwa, C. (2003). *Enterprises Take Early Lead In Web Services Integration Projects: Case Study.* Retrieved August 15, 2003, from http://www.computerworld.com/developmenttopics/development/webservices/story/0,10801,82076,00.html.

Stiroh, K. J. (June, 2001). Investing in Information Technology: Productivity Payoffs for U.S. Industries. *Current Issues in Economics and Finance.* **7** (6), pp. 1-6.

Udel, J. (2002). *Orchestrating Services.* Retrieved August 15, 2003, from http://www.infoworld.com/article/02/07/05/020708plweborch_1.html.

webMethods, Inc. (2003). *An Introduction to Web Service-Based Integration.* Retrieved August 15, 2003, from http://www.webmethods.com.

WS-I. (2003). *About Web Services Interoperability.* Retrieved August 15, 2003, from www.ws-i.org.

Young, S., Shirky, C. (2002). Are *Web Services Ready For Prime Time?* Retrieved August 15, 2003, from http://www.optimizemag.com/issue/006/squareoff.htm.

ZapThink. (2002). *The "Pros and Cons" of Web Services.* Waltham, MA: ZapThink, LLC.