# Web Services: Opportunities and Obstacles in the path of its early adoption

P. Joshi[1], H. Singh[1] and A. D. Phippen[2]

[1] School of Computing Science, Middlesex University, UK
[2] Network Research Group, University of Plymouth, United Kingdom

## Abstract

Distributed computing architecture has been around for a while but not all of its benefits could be leveraged due to issues such as inter-operability, industry standards and cost efficiency that could provide agility and transparecy to the business process integration. Web Services offer a cross platform solution that provides a wrapper around any businesses object and exposes it over the Internet as service. Web Services typically work outside of private networks, offering developers a non-proprietary route to their solutions. The growth of this technology is imminent however there are various factors that could impact its adoption rate. This paper provides an indepth analysis of various factors that could effect adoption rate of this new technology by the industry. Various advantages, pitfalls and future implications of this technology are considered with reference to a practitioner survey conducted to establish the main concerns effecting adoption rate of Web Services.

## Keywords

Internet, Web services, Evaluation

## 1. Introduction

Web Services have the potential to transform traditional distributed computing by exposing software functions or applications as services over the Internet. They use the basic Internet infrastructure to query services, publish services and to carry transactions across various services. This infrastructure allows services from distinct vendors to communicate or interact with each other. A Web Service could be a real time stock quote service, weather advisory, hotel and airline booking service or a combination of multiple such services to form an entire business process. This communication can take place through an Internet browser or any other independent application can use SOAP over Hyper Text Transfer Protocol (or HTTP) to invoke a Web Service. A service requester can query the registry for any particular Web Service, get its description and invoke the Web Service through Internet.

The growth of electronic business depends upon business-to-business (B2B), application-to-application (A2A) and business-to-consumer (B2C) interaction over the web. This requires a technology that supports inter-operability, cross platform transaction and integration of software components written in any language with legacy application. No previous distributed computing architecture (for example, Sun Java Remote Method Invocation (RMI), OMG Common Object Request Broker Architecture (CORBA), Microsoft Distributed Component Object Model (DCOM)) can deliver such benefits (Lim & Wen 2003) . All of these technologies are heavily dependent on vendor platform and tight coupling of client and server.

This paper reviews the Web Service architecture and considers the benefits and major concerns reported in literature that are faced by enterprises in implementing this technology.

Following on from this review, we evaluate these claims against a practitioner-focussed survey, which considers the literature against a sample of people who use the technologies.

## 2 A Brief Review of Web Services

There are numerous detailed technical reviews of Web Services (for example (Lim and Wen 2003, Gottschalk et. al. 2002, Walsh 2002). While we do not feel there is any benefit in providing another in this ppaer, it is worth reviewing the technologies briefly for the purposes of discussion.

Web Services are based on XML standards and facilitates writing these software services using any programming language and over any platform. Web Services programming model is based on service definition through Web Service Definition Language (WSDL) documents. These WSDL templates define the logical structure of a Web Service including the input-output parameters and reference to the location of service. Universal Description, Discovery, and Integration (UDDI) registries publish these templates. Clients across the globe can access UDDI to browse for suitable service by reading various WSDL templates. Simple Object Access Protocol (SOAP) is a communication protocol for XML based Web Services. SOAP messaging runs over HTTP, which makes it globally acceptable as most operating systems support HTTP. SOAP enables applications to communicate directly without the need for custom binaries, runtime libraries, or other platform-specific information that has plagued cross-platform data transfer in the past.

By using the Web Service architecture applications can share data or invoke methods and properties of other remote applications with out any knowledge of other application's architecture. This process is potentially expected to be much more transparent and easy to integrate in an hetrogenous environment.

Despite growth in popularity of Web Services their true commercial exploitation depends upon further development of standards in areas such as security, reliable messaging, transaction support, and workflow (Gottschalk et. al. 2002).

## 3 Factors Impacting Web Services Adoption

Web services will impact the way business objects communicate across intranet, across applications, and even across enterprises. They would provide an interface for bridging distinct technical infrastructure gaps by promoting inter-operability and flexibility. There could potentially be no more boundaries for information exchange. The joint industrial push toward standardization of Web service components has widely influenced early adoption of this technology.

If we examine the literature in the ares, we can identify a number of purported driving forces for rapid adoption of Web Services:

**Industry Support:** All major technology vendors including Microsoft, Oracle, IBM and Sun have adopted Web Service architecture. Each one of these is working to integrate support for

Web Services development. Sun is building Web Service support into its Java2 Enterprise Edition (Sun 2004) and Microsoft has built it into its .NET framework (Wolter 2001).

**Loose Coupling:** It allows web services to connect any application or data source to any target application. The Web Services world has informally adopted loose coupling because tightly coupled applications that maintains constant contact with one another do not operate well on unpredictable networks or when two applications are on opposite sides of a firewall (Shirkey 2002).

**Interoperability**: Web Services allow developers to use various programming languages, such as: Java, C++, VBScript, JavaScript, or Perl for development. Once developed they can be invoked from any platform regardless of programming language used for its development. With the use of standards-based communications methods, Web Services are virtually platform-independent (Shirkey 2002).

**Integration:** Web Services allow developers to easily integrate Business Processes across various different enterprises. They provide a much-needed interface to the software industry to integrate legacy business applications. Enterprises with distinct technical infrastructures can integrate via Web Services implementation (Shirkey 2002).

**Simplicity:** Implementation of Web Services are much simpler to implement than other distributed systems approaches. Web services help to hide back-end implementation details through the use of standardized or well-known interface definitions. The programming model for Web Services separates interfaces from implementation, which makes it easier to understand and simple to deploy (Scribner & Stiver 2002). Another added adavtage is that many of SOAP implementation tools are freely available.

**Business Expansion:** Companies would be able to access unexplored market place via UDDI public registries. Web services help companies outsource business processes. Using Web Services a company can turn over the management of a process or function such as manufacturing, logistics, or human-resources, to an outside provider (Hagel 2002).

### 3.1 Web Services Obstacles

Every technology comes with some pitfalls – it is impossible to design a loosely coupled distributed architecture, which would inter-operate with various distinct systems, without any problematic areas. Using Internet as a base technology carries along issues of security, availability, reliability and so forth. The concept of inter-operability can make any distributed system quite complex, as the entire industry needs to agree upon standards and norms. Major obstacles in the early adoption of Web Services can be identified as:

**Security:** Executing applications or calling objects over Internet could pose large problems for any IT project manager. Implementing security for transactions carried over Internet could be challenging. Security is a fundamental requirement of Web Services. Web Services could be exposing business methodologies, business applications or secure business details; this data needs to be protected from unauthorized users. Major security concerns for Web Service implementation are Confidentiality of data, Authentication or Authorization of participating parties, Integrity, Non-Repudiation and Availability of the service (Turban 2002).

**Accountability:** Or how can the service be paid for and who is responsible for the service? This has been a long debated issue with regards to payment infrastructure and access to Web Services. Either a company can have a one-time charge for accessing a web service or a periodical subscription (Ratnasingam 2002).

**Performance:** Performance is one of the major parameters to measure QoS of Web service. They can encounter performance bottlenecks due to the limitations of the underlying messaging and transport protocols (Mani & Nagarajan 2002).

## 4 Web Service Adoption and Usage Survey

While the above identifies a number of issues hampering the adoption of web services drawn from literature in the area, it has been identified in previous studies (for example, Phippen 2001, Kunda & Brooks 2000) that sometimes the reality of technology adoption differs from those reported. In particular this can be true in areas where literature is industrially driven with very little rigour or empiricism in the evidencing of analysis or results. In order to further investigate the adoption issues in the Web Service domain it was decided that a survey sampling the people that use web service technologies would greatly strengthen investigative work to date.

The survey sampled from developer groups focussing upon web service development issues. By drawing from these groups we were able to ensure developers with an interest in web service development and a good level of knowledge in the area.

Questionnaire design was based upon the claims made by the literature discussed above. There were three main areas of focus in the questionnaire:

1. Building a profile of the respondent to ascertain suitability in answering the survey and determining the depth of developer experience with web service technologies
2. The technology choices used in developing web service solutions
3. Evaluating opinion based on questions drawn from literature related to adoption issues.

### 4.1 Respondent profile

Of 100 targeted, we achieved 46 respondents, giving a response rate of 46%. The respondents represented were drawn from a number of different organisational roles: Chief Technology Officer (CTO) (45%), Project Managers (17%), IT Analysts (19%) and Others (19%) - which included Business Consultants, Technical Consultants and Software Developers. The significant response for CTOs demonstrates the strategic level of importance afforded to web services. This is not a technology used by the IT department with little impact upon the organisation as a whole.

The primary function of companies represented by the respondents was Research and Development (41%), E-Commerce/ E-Business (25%), ERP/CRM (6%) and Other (29%) including Banking, Payroll, Food Production, and Government. Respondents were fairly evenly distributed across the main coutires considered to be actively involved in web services such as USA, UK, France, Germany, Canada and Australia.

The level of of knowledge regarding web services was also encouraging when conisdering the validity of the sample set. The majority of respondents (77%) have been familiar with Web Services for over 2 years whereas 20% of them have been familiar for at least a year. Only 3%

of respondents considered themselves as beginners level. Overall the respondents provided over 100 years of experience using web service technologies.

We also determined the type of experience the respondents had with web services – were they consumers or providers. Initial results showed the over 75% of respondents were already consuming web services and over 75% were also developing web services. An interesting results came from comparing those consuming web services against those developing their own, detailed in table 4.1. Almost 100% of those respondents who consumed web services were also developing their own.

| | | Planning to develop Web Services | | | | |
|---|---|---|---|---|---|---|
| | | Already started | Testing various | Thinking about it | No idea | Total |
| Planning to use Web Services | Already | 32 | 2 | 0 | 0 | 34 |
| | Considering | 1 | 1 | 1 | 0 | 3 |
| | Don't know | 1 | 1 | 1 | 6 | 9 |
| Total | | 34 | 4 | 2 | 6 | 46 |

**Table 4.1. Planning to use Web Services vs Planning to develop Web Services**

## 4.2. Technology Choice

While the argument for web services is that they are essentially technology neutral the fact remains that practitioners have to make a choice regard which technology to adopt to develop their own web service strategy. Two dominant approaches to building Web Services applications are either to go with Microsoft's .NET architecture or the rest of the industry's java based approaches. Coupled with this choise is the option of development tools. While Microsoft, unsurprisingly packages their web service technology in their Visual Studio .Net the options are more diverse for J2EE. A Number of companies such as IBM, BEA, Cape Clear and Oracle has built some or all of their Web Services offerings on top of J2EE.

| Platform | % respondents |
|---|---|
| Sun J2EE | 49% |
| Microsoft .Net | 41.2% |
| Other | 9.8% |

| Development technology | % respondents |
|---|---|
| Open source | 29.0 |
| Sun Microsystems | 17.4 |
| Microsoft | 23.2 |
| BEA | 11.6 |
| IBM | 7.2 |
| HP | 2.9 |
| CapeClear | 2.9 |
| Other | 5.8 |

**Table 4.2. Platform and development technology choices**

The responses from our survey do not give too many surprises when examined independently. There is a fairly even split between Sun and Microsoft technologies and a few "other" responses (most of those use Perl's web service technologies). Again, with the choice of development technology there is a reasonably straightforward split. However, the "open source" response is more interesting when compared against the choice of platform in cross-tabular analysis (see table 4.3). The are a significant number of developers using open source technologies to develop .Net solutions. This certainly highlights the standards focussed approach to web services. Literature states that it *should* make no difference which application server platform is used to develop these services (Clabby 2003), and our survey indicates that this may certainly be the case.

| **% of cases** | | **Development Technologies** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Open Source | Sun | Microsoft | BEA | IBM | HP | CapeClear | Other |
| **Platform Choice** | Sun J2EE | 40.5% | 29.7% | 18.9% | 21.6% | 13.5% | 5.4% | 2.7% | 8.1% |
| | Microsoft .Net | 29.7% | 10.8% | 40.5% | 10.8% | 8.1% | 2.7% | 2.7% | 5.4% |
| | Other | 10.8% | 10.8% | 10.8% | 5.4% | 5.4% | 2.7% | 0 | 2.7% |

**Table 4.3. Platform choices vs. development choices**

### 4.3. Evaluation of adoption issues

When evaluating adoption issues we decomposed the benefits and barriers drawn from the literature into 3 broad areas:

- standards – generally viewed as one of the main reasons to adoption web services and provide numerous benefits to interoperability and integration;
- complexity – or lack thereof. The literature suggests that web service adoption is possible due to the simplicity of development and integration. This certainly differs from findings relating to previous distributed software technologies (Phippen 2001, Kunda & Brooks 2000). However, this simplicity also hampers performance and perhaps reliability;
- security – possibly the biggest barrier to web service adoption and distributed, loosely coupled systems in general. Issues such as accountability and reliability also draw upon this security domain.

The questionnaire focussed upon these three core issues in presenting a number of statements and requesting closed, bipolar responses (strongly disagree, disagree, no opinion, agree, strongly agree). These responses were coded from 1 to 5 in order to ascertain mean response, standard deviation, and skewness of data. The statements presented were:

a. Web services have the potential to transform traditional distributed computing
b. Web services are easy to code and deploy compared to other distributed solutions such as CORBA and DCOM
c. Web services offer interoperability between applications by supports code development using any software language
d. Support from major technology vendors such as Microsoft, Sun, IBM and HP is a basic reason for web service adoption
e. Web services allow developers to easily integrate business processes across various enterprises
f. There are enough security standards available for web services
g. Adoption rates for web services would improve with further development of security standards

The core responses are detailed and discussed below:

|  | *a* | *b* | *c* | *d* | *e* | *f* | *g* |
|---|---|---|---|---|---|---|---|
| **Mean** | 3.96 | 3.93 | 3.93 | 3.46 | 3.74 | 2.80 | 3.76 |
| **Std. Deviation** | .868 | 1.020 | .975 | 1.130 | .976 | 1.222 | .993 |
| **Skewness** | -.766 | -.522 | -1.368 | -.516 | -.491 | .239 | -.630 |

**Table 4.4. Core statistics for adoption issues**

|  | *a* | *b* | *c* | *d* | *e* | *f* | *g* |
|---|---|---|---|---|---|---|---|
| **Strongly disagree** | 0% | 0% | 4.3% | 4.3% | 0% | 13% | 2.2% |
| **Disagree** | 8.7% | 10.9% | 4.3% | 21.7% | 15.2% | 37% | 8.7% |
| **No opinion** | 13% | 21.7% | 10.9% | 13% | 17.4% | 15.2% | 23.9% |
| **Agree** | 52.2% | 30.4% | 54.3% | 45.7% | 45.7% | 26.1% | 41.3% |
| **Strongly Agree** | 26.1% | 37% | 26.1% | 15.2% | 21.7% | 8.7% | 23.9% |

**Table 4.5. Response outcomes for adoption issues**

Results show a positive response for complexity issues (*statements a-c*) with the web service domain. In particular issues such as the potential of web services to realise distributed computing goals, and the use of web services to achieve interoperability are very positively represented. One point of interest is the level of "no opinion" on statement b. Given the responses for other issues surrounding development complexity in web services, we put forward the assumption that respondents perhaps did not have experience using older distributed software technologies such as CORBA and DCOM.

However, it seems from the results the we can confirm that the development of distributed software solutions from web service approaches does offer significant potential without the complex knowledge requirements traditionally associated with this type of development (Brown & Wallnau 1996)

The issue of standards is less clear from the immediate results. *Statement d* in particular does not show such a strong positive response (although its mean value and skewness both veer toward an overall positive response). As this is put forward in literature as one of the most positive aspects to web services, this is an interesting outcome. Further cross table analysis also fails to identify any obvious trends in developer feeling on this issue – open source developers are split fairly equally on whether it is a positive or negative issue, and those who view business process integration as an important issue also don't show any trends.

The one area where it was very apparent that participants were not confident regarding web services was the issue of security standards. The majority of respondents believed that adoption rate of Web Services would drastically improve with further development of security standards. This was as expected since security has already been identified as a major threat to growth of Web Services. In order to develop the issues surrounding security further, we posed a final question: what was the most important issue for Web Service security. Table 4.6 details the outcomes:

| | Response % |
|---|---|
| **Authentication** | 43.5 |
| **Confidentiality** | 28.3 |
| **Integrity** | 8.7 |
| **Non repudiation** | 2.2 |
| **Other** | 17.4 |

**Table 4.6. Most important security issue affecting web services**

## 7. Discussion and Conclusion

Web Services represent a technology migration from vendor specific software development to Internet based cross platform and vendor neutral environment. The use of Simple Object Access Protocol (or SOAP) over HTTP for communication makes it easily available over Internet. Most software companies are already investing their resources into developing or implementing Web Services environment. For others it's a matter of time before which they would have to make a decision to jump onto the bandwagon. Web services are worth paying attention even for organisations that don't plan an immediate launch.

Literature has stated that Web Services offer low transaction cost, improved ROI, extensibility of legacy software and moves business process over to Internet. If the potential express in literature could be realised in reality the flexibility and power afforded by such approaches could offer major advantages to a company.

The online survey established Web Services as solution provider for distributed software development and interoperability and, to a slightly lesser extent, business integrity. However, one major issue arising from our study is the non-availability of security standards for Web Services. Although securit concerns still remains high on agenda of companies looking to adopt Web Services. Authentication standards need to be developed further to support user as well as device authentication model. This would not only provider higher level of authentication but also support better accountability management solution in distributed computing architecture. This would lead to cost efficiency and resolve issues of Non-Repudiation.

However, condisering the results from our survey the adoption rate of Web Services is positive and in the coming years further developments would be made in this field. It is arguably the first distributed software technology whose complexity will not hamper its development. Looking at the progress made by Web Services since its start few years back and considering the benefits offered by them to the business world it would seem that Web Services will have to positive impact in the future.

## 8. References

Brown & Wallnau (1996). "A Framework for Evaluating Software Technology". IEEE Software September 1996.

Clabby, J. (2003), "Web Services Explaind:Solution and Applications for the Real World", publication: Prentice Hall.

Gottschalk K. et. al (2002), " Introduction to Web Services Architecture", IBM System Journal 41, No. 2.

Hagel, J. (2002), ”Edging into Web Services ”, McKinsey Quarterly, Special Edition: Technology Issue 4, p29, 9p, 3bw.

Kunda & Brooks (2000). “Assessing Obstacles to Component Based Development: A Case Study Approach”. Information and Software Technology vol. 42.

Lim B, Wen, H. (2003), “Web Services: An Analysis of the technology, Its Benefits, and Implementation Difficulties”, Information System Journal, Vol. 20 Issue 2, p49, 9p; (*AN 9147819*)

Mani, A., Nagarajan, A. (2002),"Understanding Quality of service for Web services: Improving the performance of your Web services", http://www-106.ibm.com/developerworks/library/ws-quality.html

Phippen, A. (2001). "Practitioner Perception of Component Technologies". Proceedings of Euromedia. Society for Computer Simulation.

Ratnasingam, P. (2002), “The importance of technology trust in Web services security”, Information Management & Computer Security 10/5 255±260

Scribner, K., Stiver, M (2002), “Applied SOAP: Implementing .NET XML Web Services“, publication: SAMS Publication.

Shirky, C. (2002), ”Planning for Web Services: Obstacles & Opportunities”, O’Reilly.

Sun (2004), “Java 2 Platform, Enterprise Edition (J2EE)” http://java.sun.com/j2ee/

Turban, E. (2002), “Electronic Commerce: A Managerial Perspective”, Prentice Hall Publication.

Walsh, A. (2002), "UDDI, SOAP, and WSDL: The Web Services Specification Reference Book", Pearson Education Asia.

Wolter, R. (2001), “XML Web Services Basics”, http://msdn.microsoft.com/library/?url=/library/en-us/dnwebsrv/html/webservbasics.asp?frame=true