A Comprehensive Evaluation Of The Key Factors Accounting For The Efficacy Of Intelligent Web Systems[1]

Janhvi Garg

Department of Information Technology, Bharati Vidyapeeth College of Engineering
Affiliated to Guru Gobind Singh Indraprastha University, New Delhi

DOI:10.37648/ijrst.v12i01.008

Received: 17 February 2022; Accepted: 20 March 2022; Published: 28 March 2022

ABSTRACT

The grand vision of Tim Berners-Lee, director of the World Wide Web Consortium (W3C) founded in 1994, of changing the non-semantic Web (Web 1.0, Web 2.0) to a semantic Web (Web 3.0) will connect all websites and make their systems interoperable. Though this system has not yet fully matured, the goal of utilizing the full potential of the web by creating an interoperable knowledge whole is not far from reach. With the emergence of various web technologies and innovative concepts of using the web to its fullest potential, the web is evolving rapidly toward intelligent web systems. Ideally, intelligent web systems will be a combination of a semantic web and various web services where computers can automatically process web contents and integrate their services. This paper reports on which web technologies and protocols succeeded in realizing the current web and speculates on the possible future web architecture and its social impact.

THE SUCCESS STORY OF THE WEB

In 1989, Tim Berners-Lee [1] invented the World Wide Web (WWW or W3), an internet-based hypertext transfer protocol (HTTP) for global information sharing. Hypertext Markup Language (HTML) was developed for displaying web pages on browsers like Netscape or Internet Explorer running on a user's computer. Domain Name Service (DNS) and Universal Resource Locator (URL) naming conventions were developed to discover a machine on the internet hosting a web page.

Since the inception of the web, the explosive growth in the number of websites and web users is a clear indication of its popularity among students, researchers, businesses, and shoppers alike. The web is the most efficient medium for information collection and dissemination in the fastest and cheapest possible way. It has changed everyday life, transforming how students study, teachers teach, and companies operate. This explosive growth of the web has also been facilitated by the increased popularity of personal computers (PCs) and faster internet access speeds.

Despite the web's huge success as a platform and its widespread social impact, the web has not been studied and understood well as a whole system. This paper explores the web technologies and tools enabling certain web applications and social interactions. These technologies have facilitated social interactions that, in turn, are shaping future application needs and infrastructure requirements (see Figure 1) [2]. To engineer the future web, it is essential to study and understand the complex dynamics of social interactions, application needs, and infrastructure requirements driving web development.

---

[1] *How to cite the article:*
Garg J., A Comprehensive Evaluation Of The Key Factors Accounting For The Efficacy Of Intelligent Web Systems, IJRST, Jan-Mar 2022, Vol 12, Issue 1, 53-56, DOI: http://doi.org/10.37648/ijrst.v12i01.008

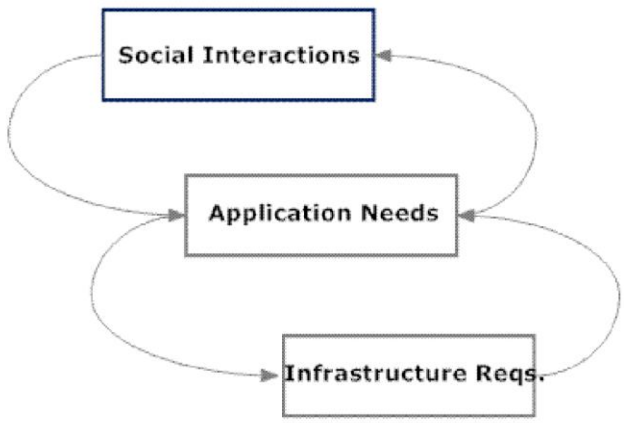

Figure 1. The social interactions enabled by the web put demands on the web applications behind them, in turn putting further demands on the web's infrastructure [2].

FUTURE WEB: THE INTELLIGENT WEB SYSTEMS

By adding semantics to web documents, computers can browse, search, and query on behalf of human users. Intelligent agents can be designed and employed in applications that group information distributed throughout the web, process it, and take action accordingly. Interoperability is possible only with an efficient and flexible middleware design using semantic web technologies, which provide a standard set of interfaces for heterogeneous applications to communicate.

What Semantic Web Enables

Many internet languages have been designed using XML to add structure and semantics to web content. These include Wireless Application Protocol (WAP) for mobile devices, Rich Site Summary (RSS) for aggregating content from multiple websites, Synchronized Multimedia Integration Language (SMIL), Resource Description Framework (RDF) for interpreting metadata uniformly across machines for application interoperability, and Web Service Description Language (WSDL).

RDF is seen as a framework for realizing the full potential of XML by interpreting metadata meaning uniformly between applications or by software agents. XML and RDF describe and structure web data, enabling it to be queried by other web documents. Non-XML data exchange formats like N3, Turtle, NTriples, and Web Ontology Language (OWL) have also emerged, setting design principles for adding semantics to web content to ensure interoperability between applications.

While many technologies compete, the technology that is less verbose, easy to use, and flexible enough to support distributed computing environments will ultimately dominate.

With the ability to transform the non-semantic web into a semantic web and make it interoperable, many innovative uses of the web have been conceptualized. New web service concepts and business models are emerging. One such model is Software as a Service (SaaS), where an application is hosted as a service and accessed over the internet by clients. The SaaS [3] business model will change how people build, sell, buy, and use software. In this model, the application architecture and delivery architecture are separate layers, providing better flexibility and adaptability to changing business models.

SaaS offers several benefits. It eliminates ownership and maintainability challenges for clients. A single hosted service can cater to multiple clients, and as it scales, the operating cost per client decreases. Since a single application instance on a server can cater to multiple clients, the application can have different appearances and behaviors for each client, configurable through a metadata service [3].

It is a proven fact that the success of a business [4] depends on its visibility and communication with prospective clients. Advances in integrating technologies to create intelligent and flexible middleware have led to the emergence of Service-Oriented Architecture (SOA) (see Figure 2). SOA allows businesses to register services and deliver the SaaS business model [4].

SOA has several benefits, including supporting automatic discovery of services. Instead of building monolithic applications, service providers can use each other's services to provide standard means of software application interoperation across platforms. These intelligent web systems will ideally combine the semantic web and various web services.

The next section explores the web service architecture designed to facilitate automatic service discovery and integration of services in the SOA architecture.

Web Services Architecture and Technologies

Web services can be simple web APIs (Application Processing Interfaces) on client machines for accessing applications hosted on remote servers over the internet, where the actual execution of the requested service takes place. In SOA architecture, clients and service providers communicate using XML messages, which in turn utilize the Simple Object Access Protocol (SOAP). SOAP defines message constructs for exchanging XML messages between services.

Compared to other distributed object models that were popular in the past, such as DCOM (Distributed Component Object Model) and CORBA (Common Object Request Broker Architecture), SOAP is platform-independent. Because SOAP uses XML, it is not tied to any specific programming language or operating system, making SOAP-based web services open and accessible to all. The decline of CORBA [5] is attributed to the complexity of its object model and language mapping mechanisms. Microsoft's DCOM also failed as middleware because it supported only the Windows operating system.

When the service provider is known a priori to the client, this mode of service access is similar to a simple Remote Procedure Call (RPC). For automatic discovery of services, service providers register their service metadata—such as the provider's name, type of service, service policy, and contracts—with a service broker, as mentioned in Section 3.1 [4]. Universal Description, Discovery, and Integration (UDDI) [6] is a platform-independent, XML-based registry for businesses worldwide to list their services on the internet. UDDI provides standard mechanisms for dynamic service discovery, similar to how Domain Name Service (DNS) registers web addresses for automatic discovery. Service providers advertise their services in Web Service Description Language (WSDL).

Figure 2 illustrates the service-oriented architecture model for service interoperability between providers A and B. Service requests from A are wrapped with SOAP before being sent to B, and the services provided by B are similarly wrapped with SOAP before being delivered to A. Service provider A becomes aware of service provider B because B registered its business with UDDI and advertised its services using WSDL. Likewise, service provider B can request services from A because A has also registered its services with UDDI.

The on-demand, automatic discovery of services enables the development of advanced intelligent applications. For instance, consider an online flight booking system. Almost all airlines host electronic ticket booking services on the internet individually. A travel agency service can integrate these airline services by discovering them in real-time to compare ticket prices and present them to its own customers. Many Fortune 500 companies and large enterprises are gravitating toward these on-demand service solutions [7].

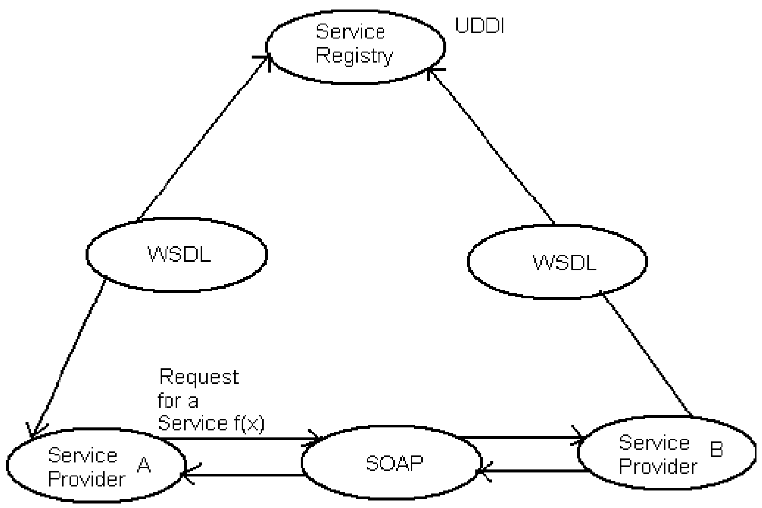

Figure 2: Service Oriented Architecture for Interoperability of Services

(IJRST) 2022, Vol. No. 12, Issue No. I, Jan-Mar **e-ISSN: 2249-0604, p-ISSN: 2454-180X**

DISTRIBUTED OR GRID COMPUTING

Distributed or Grid computing [8], involving heterogeneous computing resources from geographically dispersed areas connected with fast internet connections, is employed for processing loosely coupled parallel tasks of large-scale, high-performance engineering and science applications. This approach will benefit from the semantic and SOA architecture of the web. The heterogeneous computing resources of Grid computing require middleware based on semantic web technologies like annotating data with XML and RDF, Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI). These technologies provide a standard set of interfaces for communication between heterogeneous computers. Thus, Grid computing can be seen as a semantic web application, and as semantic web technologies mature, Grid computing will benefit even more.

In the near future, intelligent web systems are expected to bring many non-computing devices, such as TVs, phones, and medical instruments, online with novel service offerings. These web services will deliver scalable, flexible, interoperable, cost-effective, and high-performance solutions catering to a larger human society.

CONCLUSION: CREATING THE FUTURE

The future of the web and the future of human society are intermingled. To plan a better future, web science needs to be studied and understood as a whole. Web page designers and application developers face numerous challenges in supporting high-performance infrastructure connecting multiple servers and services distributed across the globe. They must adhere to standardization and layered design techniques to ensure reusability, interoperability, and scalability, allowing for the web systems' future growth.

Many new ideas for using the web need to be conceptualized to help the web evolve and serve a larger global audience. Some of these ideas can be realized with current web technologies, while others will necessitate the development of new design principles, standards, web technologies, and tools.

REFERENCES

[1] Tim Barners-Lee. *www.w3.org/People/Berners-Lee/*. August 2008.

[2] J. Handler, N. Shadbolt, W. Hall, T. Berners-Lee and D. Weitzner. Web Science: An Interdisciplinary Approach to Understanding the Web. *Communications of the ACM*, Vol 51, No. 7, July 2008.

[3] F. Chong and G. Carraro. Architecture Strategies for Catching the Long Tail. *Microsoft Corporation*, April 2006.

[4] J. Banerjee and S. Aziz. SOA: The missing link between Enterprise Architecture and Solution Architecture. *SETLabs briefing*, Vol 5, No 2, pp 69-80, March 2007.

[5] M. Henning. The Rise and Fall of CORBA. *Communications of the ACM*, Vol. 51, No. 8, August 2008.

[6] *http://en.wikipedia.org/wiki/web_service*, August 2008.

[7] Reuters Business and Finance. *www.reuters.com/articles/pressrelease.* May 2008.

[8] C. Globe, D. DE. Roure. Semantic Web and Grid Computing. *www.semantic grid.org/documents/swgc/swgcfinal. pdf*, August 2008.