All Sciences Proceedings
http://as-proceeding.com/

5th International Conference on Applied Engineering and Natural Sciences

July 10-12, 2023 : Konya, Turkey

https://www.icaens.com/

Developing Web Services and Challenges

Faris Sattar Hadi

Information Technology Research and Development Center, University of KUFA, Iraq

fariss.alkaabi@uokufa.edu.iq

Abstract – Web services have recently sparked a lot of attention among vendors and researchers. The issue of application integration may have a flexible solution thanks to web services, which are based on open standards and already-existing Internet protocols.Web services are becoming more and more common in web applications with the use of WSDL, SOAP, and UDDI. The present Web services designs, though presented with certain obstinate issues, like security. We will provide an outline of these issues in this essay. We think that the success of Web services will depend on resolving these issues. Finally, we anticipate distinct developments in semantic Grid services.

Keywords – Web Services, Web Services Composition ,Grid Services, Simple Object Access Protocol, Web Services Description Language.

INTRODUCTION

Web service is essentially an independent piece of software with a specific URI (the Uniform Resource Identifier is a specific address), technology that utilizes the Internet, particularly the Web. Web services have a supplier, as well as (ideally) consumers or subscribers, according to the fundamental concept. The service provider must not only develop the service but also produce a specification and publish it in a directory so that potential clients can access it. Potential users can search the service directory to find available services. In order to discover a single service that meets their needs, or a group of services that can be combined effectively, service clients or users will often search through such a directory.

It is obvious that Web services must be interoperable. Additionally, they must run on every Web browser and be independent of the operating system that they are installed on.

Regardless of their programming languages, service engines should be able to communicate with one another. Web services are frequently built on top of standards to accomplish these goals; the most popular ones at the moment are the XML-based specifications SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery and Integration), and WSDL (Web Services Description Language), though this situation is constantly changing . The following are the fundamental steps of publishing and using a software component that has been developed as a Web service:

A UDDI directory, which serves as a central repository of metadata for all registered services and keeps data about them, is where the provider of a Web service registers his service.

Examples of this meta data include the author, the service category, and technical details. Since UDDI data is often kept on a number of distant servers, UDDI also defines a query language, authoring authorisation, and replication method. This guarantees that a service can be quickly located in response to a request; a request is satisfied by providing instructions on how to access and use the service at the provider's server.

A WSDL file, which is not a part of the UDDI directory, contains the guidelines for client and Web service communication.

Rather than referred to by the details a client requests from the UDDI directory. This document outlines the requirements for the input and output data as well as the communication protocol. In order to send and receive SOAP messages with the actual Web service, a proxy is created using the WSDL specification [1, 2].

Ad hoc methods have been employed in business-to-business applications in recent years to benefit from the fundamental Internet architecture.

Recently,built on top of current Web protocols and open XML standards, web services are emerging as a methodical and extendable framework for application-to-application interaction.

A new type of Web application is called a web service.These apps are self-contained, self-describing, and modular and may be published, found, and used.

All around the Web services can be used to create and carry out complex business operations or just make straightforward information requests.

After being deployed, a Web service can be found and used by other applications or Web services.

The ability to build applications instantly utilizing loosely linked, reusable software components is the main benefit of using Web services.

This has important ramifications for both technological and commercial uses. Instead of being sold as packaged goods, software might be delivered and paid for as a continuous stream of services. To complete business tasks, it is possible to achieve automatic and dynamic interoperability between systems. The Internet allows for entirely decentralized distribution of business services that can be accessed by a large instead put their attention on the value of their services and other crucial duties rather than the burden of complex, expensive, and low-quality software integration.

The Internet followed will develop into a universal platform for communication between businesses and individuals in order to conduct various commercial operations and offer value-added services. To make new offerings and markets more accessible to small and medium-sized businesses, the barriers to entry will be decreased. Dynamic businesses and value chains become possible and perhaps even necessary for competitive advantages [3].Show Fig.(1).

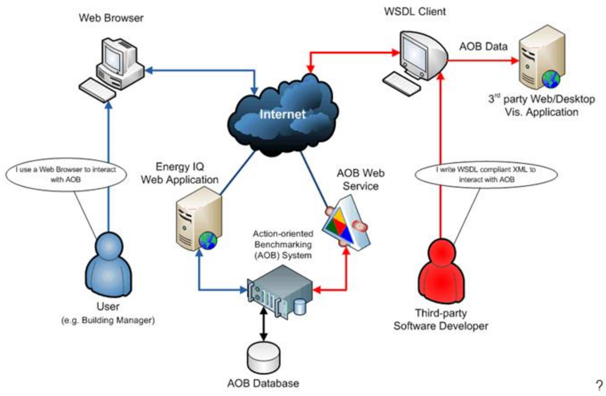

Fig.(1): Web Services Map.

WEB SERVICES DIFFICULTIES AND CHALLENGES

Technologies like SOAP, WSDL, and UDDI are crucial for enabling Web services. But in order to properly meet the needs of business applications, the present Technologies are not perfect. The three main issues and possible research trajectories to improve the current technology are covered in this section.

Let's utilize a straightforward trip scenario to highlight the security issue with Web services. The Web services framework requires more than three components should appropriately interrelate in order to finish the travel scenario.

We must, at the absolute least, guarantee that operations like electronic check-ins were carried out in a safe setting and that messages were reliably transmitted to the intended locations. Why do we need to provide more security when technologies like Kerberos, HTTP secure (HTTPS), and secure multipurpose internet mail extensions (S-MIME) are already available? The distinction between end-to-end and single-hop usage holds the key to the solution. Business messages frequently start out in one program and then move to another.

In order to secure (for confidentiality) a direct connection between two machines, mechanisms like secure sockets layer are excellent.

No assistance is provided if the message must cross multiple connections.

Fundamentally, the security issues that could impact Web services are the same ones that have troubled traditional Web-based systems.

Numerous of these were thoroughly covered in [4,5,6,7,8,9,10]. Here, we offer the following summary of the current scenario: Enterprise adoption of Web services depends on security, yet the Web services architecture does not currently adhere to fundamental security standards.

Web services require message exchange, hence protecting the message exchange is a crucial topic to take into account when developing and utilizing Web services.

Security in the context of Web services refers to the ability of the message's recipient to confirm the message's integrity and ensure that it has not been altered. The communication should have been delivered in confidence to the recipient, who should be aware of the sender's identity and whether the center is permitted to do the operation requested in the message. Typically, messages are encrypted to meet these requirements [11].

However, because Web services enable communication between all systems, internal and external, over HTTP ports, the application servers are invariably attack surface that is "application level" open.

A few standards, such as WS-security and numerous additional initiatives (mainly from the major manufacturers and PKI providers) towards providing digital signatures on XML messages and transactions, have been released to address the message security issue. But "application level" attacks barely raised an issue.

The application servers are invariably HTTP-based due to Web services' ability to facilitate communication between all systems, internal and external, through these ports open "application level" attack surface To solve the message security issue, a few standards, like WS-security, and countless further initiatives (mostly from the major manufacturers and PKI suppliers) toward enabling digital signatures on XML messages and transactions have been developed. "Application level" attacks.

The OASIS standard Security Assertion Markup Language (SAML) [12] gives partners programs a way to exchange user authentication and permission data.

In essence, all significant vendors' e-commerce products offer this single sign-on (SSO) feature. In the lack of a standardized protocol for communicating authentication data, companies typically implement SSO via cookies in HTTP communication. With the development of SAML, the same data can be wrapped in XML in a standard manner, negating the need for cookies and enabling interoperable SSO.

Composition Difficulties and Solutions

Support for greater levels of business functionality is needed for complex business interactions. Typically, business transactions involve lengthy execution processes and involve a variety of relationship interactions. We need to be able to express business processes and service states as well as develop service compositions (complex aggregations) in a consistent and methodical way in order to deploy and use these types of services successfully. There are numerous solutions proposed to carry out this task; for examples, see Web Services Flow Language [13], XLANG [14], and BPEL4WS [15].Show Fig.(2).

Various terminologies have been used in the industry to indicate how components might be connected to construct intricate corporate procedures. Systems for managing workflow and document movement between diverse resources in an IT organization have been around for a while. These resources often include some human interaction and can be people, systems, or applications. Business process management systems (BPMS) have also been employed to help organizations create top-down process design models that include a variety of integration tasks (such as integration to legacy systems). A business process's entire lifespan, including the modeling, executing, monitoring, management, and optimization responsibilities, would normally be covered by BPMS systems [16,17].

The composition of Web services in a process is now described by words like "Web services composition" and "Web services flow," among others.

Flow The words orchestration and choreography have also been used to describe this in more recent times.

The term "orchestration" refers to the process of describing how Web services can communicate with one another at the message level, including the business logic and sequencing of the interactions.

These interactions, which may involve several apps, companies, or both, would produce a long-lasting, transactional, multi-step process model.

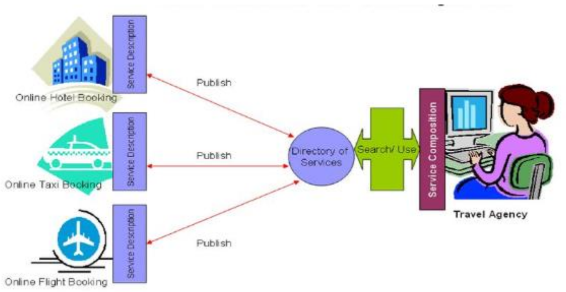

Fig.(2): Web Services Composition Sample.

SOLUTION FOR SOME DIFFICULTIES AND CHALLENGES

The semantic component is still missing from the existing Web services technology, which mostly provides a syntactical answer. In WSDL, a Web service is described, outlining what data the service requires as input and what results it produces. Web services must have the ability to orchestrate themselves into more complicated services in order to fully realize their potentials (beyond the business application connection).

Therefore, we require techniques for fusing distinct Web services to create distributed, higher-level services.

The first steps have been taken by the Web Service Flow Language (WSFL), which may represent the sequencing of separate services. The user can choose which Web services to combine and in what order using WSFL. To enable software agents to find, recognize, and combine services, we still need a framework that semantically characterizes services.

Many scholars think that the Semantic Web concept of the next-generation Web, which allows computers to comprehend Web material without any ambiguity focuses specifically on this issue [18, 19, 20]. Tim Berners-Lee is the creator of the Semantic Web project, which aims to build a web that can be processed by machines. The Web community's more research-focused individuals are the main supporters of the semantic web.

On the other side, industrial players like Microsoft, IBM, and BEA have heavily influenced the development of Web Services due to business interests.

The director of the World Wide Web Consortium outlined how to make the two work together in his opening address at the Twelfth International WorldWideWeb conference.

The primary focuses of Web development cooperate rather than compete. Web Services, according to Berners-Lee, address urgent technology needs, while the The Semantic Web could experience exponential expansion in the future.

GRID SERVICES

A robust computing infrastructure is required to handle complex Web service applications. This infrastructure is provided by grid computing. It's Open An progression toward a grid system architecture based on Web services concepts and technologies is represented by the Grid Services Architecture (OGSA) [21] .Show Fig.(3).

The Open Grid Services Architecture (OGSA) builds a distributed system framework on top of the Open Grid Services Infrastructure (OGSI) [22] by integrating essential Grid technologies with Web services mechanisms. A Grid service instance is a service that complies with a set of standards for lifespan management, character discovery, and notification. These standards are described as Web Service Definition Language (WSDL) interfaces, extensions, and behaviors. The regulated management of distributed, frequently long-lived services is made possible by grid services and is frequently needed in complex distributed applications.

The standard factory and registration interfaces for developing and locating Grid services are also introduced by OGSI.

Using a Grid Service Handle (GSH) and a Grid Service Reference (GSR), client applications (perhaps located remotely) can access grid service instances [21,23]. You can think of a Grid Service Handle as a persistent network pointer to a specific Grid service instance. The GSH doesn't offer enough details for a client to be able to access the service instance. In order to "resolve" a GSH into a Grid Service Reference, the client is required. All the information required to reach the service instance is contained in the GSR.

Using a Grid Service Reference, a client application can send requests directly to a certain instance at a given network-attached service end point.

Grid Service Reference-identified.In order to support client resolution of a Grid Service Handle into a Grid Service Reference, OGSI offers the HandleResolver[22]. Furthermore, the specific service provider-side implementation architecture is not mandated by OGSI.

How OGSI interfaces are likely to be used by client programs is a crucial additional concern. OGSI takes advantage of a key feature of the Web services architecture, which is the use of WSDL to specify various protocol bindings, encoding schemes, messaging schemes, and other aspects for a given Web service.

Recently, many international conferences, including SC2002 (the international conference for high performance computing), have featured grid services Grid2002, and communications), etc.

Semantic Grid services are an additional intriguing topic [24]. The Grid's current state of development is similar to the Web's a few years ago. there is a small deployment, primarily driven by enthusiasts in the scientific community, with growing standards and a little amount of commercial uptake. The Semantic Web could also be compared to this. In the meantime, machine-to-machine communication (XML) on the Web has replaced machine-to-human communication (HTML). This is the precise infrastructure that the Grid requires. The assumption that Grid adoption will follow the same exponential development pattern as the Web is alluringly drawn from these parallels.

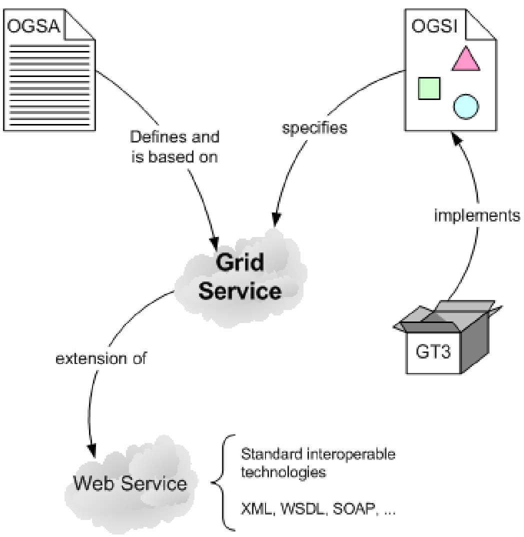

Fig.(3): Grid service.

CONCLUSION

We have discussed Web services difficulties and challenges as a new Web technology, in this research Web services were discussed from different perspectives such as semantics and service composition. They are essential for the deployment of Web services to be successful and developed.

We conclude by providing a glimpse of Grid services. We think that the integration of Grid and Semantic Web services will be a significant technical development.

REFERENCES

[1] Keidl, M., Kreutz, A., Kemper, A., Kossmann, D. (2002). A Publish & Subscribe Architecture for Distributed Metadata Management. In Proc. 18th IEEE Int. Conf. on Data Engineering (ICDE), San Jose, CA, February 2002. IEEE Computer Society, pp. 309–320.

[2] Pilioura, T., A. Tsalgatidou (2001). E-Services: Current Technology and Open Issues. In Proc. 2nd Int. Workshop on Technologies for E-Services, Rome, Springer-Verlag, Berlin, pp. 1–15.

[3] D. Fensel, C. Bussle, Web services modeling framework, Electron. Commerce Res. Appl. 1 (2002) 113–137.

[4] D. Farber, Balancing security and liberty, IEEE Internet Comput. 5 (6) (2001) 96–96.

[5] G. Goth, Securing the internet against attack, IEEE Internet Comput. 7 (1) (2003) 8–10.

[6] A. Householder, K. Houle, C. Dougherty, Computer attacktrends challenge Internet security, IEEE Comput. 35 (4) (2002) 5–7.

[7] G. McGraw, Managing software security risks, IEEE Comput. 35 (4) (2002) 99–101.

[8] W.T. Polk, N. E Hastings, A. Malpani, Public key infrastructures that satisfy security goals, IEEE Internet Comput. 7 (4) (2003) 60–67.

[9] D. Scott, R. Sharp, Abstracting application-level web security, in: Proceedings of the Eleventh International World Wide Web ConferencE (WWW), Honolulu, Hawaii, USA, 7–11 May 2002.

[10] D. Scott, R. Sharp, Developing secure web applications, IEEE Internet Comput. 6 (6) (2002) 38–45.

[11] E. Wales, Web services security, Comput. Fraud Security 18 (1) (2003) 15–17.

[12] http://www.oasis-open.org/committees/security/.

[13] http://www3.ibm.com/software/solutions/webservices/pdf/ WSFL.pdf.

[14] S. Thatte, XLANG-Web Services for Business Process Design, http://www.gotdotnet.com/team/xml wsspecs/xlangc/ default.htm.

[15] http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/.

[16] L. Juhnyoung, J. Yang, J. Chung, Winslow: A Business Process Management System with Web Services. IBM Research Report, November 2002.

[17] L. Frank, D. Roller, M. Schmidt, Web services and business process management, IBM Syst. J. 41 (2) (2002) 198–212.

[18] N. Gibbins, S. Harris, N. Shadbolt, Agent-based Semantic Web Services, WWW 2003, Budapest, Hungary, 20–24 May 2003.
[19] J. Hendler, Agents and the semantic web, IEEE Intell. Syst. 16 (2) (2001) 30–37.
[20] S. McIlraith, T.C. Son, H. Zeng, Semantic web services, IEEE Intell. Syst. 16 (2) (2001) 46–53.
[21] I. Foster, C. Kesselman, J. Nick, S. Tuecke, The physiology of the grid: an open grid services architecture for distributed systems integration. Open Grid Service Infrastructure WG, Global Grid Forum, June222002.http://www.globus.org/research/papers/ogs a.pdf.
[22] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman et al. (Eds.), Open Grid Services Infrastructure(OGSI), 2003, http://www-unix.globus.org/toolkit/draft-ggfogsi- gridservice-33 2003-06-27.pdf.
[23] I. Foster, C. Kesselman, J. Nick, S. Tuecke, Grid services for distributed system integration, IEEE Comput. 35 (6) (2002)37–46.
[24] D. De Roure, N.R. Jennings, N.R. Shadbolt, The Semantic Grid: A Future e-Science Infrastructure, 2003,http://www.semanticgrid.org.